Máster Título Propio

Creación de Interfaces y Aplicaciones en Red

tech universidad FUNDEPOS

tech universidad FUNDEPOS

Máster Título Propio
Creación de Interfaces y Aplicaciones en Red

- Modalidad: **online**
- Duración: **12 meses**
- Titulación: **TECH Universidad FUNDEPOS**
- Dedicación: **16h/semana**
- Horario: **a tu ritmo**
- Exámenes: **online**

Acceso web: www.techtitute.com/informatica/master/master-creacion-interfaces-aplicaciones-red

Índice

01
Presentación
pág. 4

02
Objetivos
pág. 8

03
Competencias
pág. 14

04
Estructura y contenido
pág. 18

05
Metodología
pág. 32

06
Titulación
pág. 40

01

Presentación

Intervenir de forma competitiva en el ámbito de la computación requiere de unos sólidos conocimientos que incluyan las novedades y actualizaciones más relevantes en ingeniería del software, bases matemáticas, estadísticas y otras áreas. En este programa hemos construido un camino intensivo de especialización a través del cual, el profesional podrá capacitarse científica y tecnológicamente en ingeniería informática, con calidad, seguridad y un objetivo último de excelencia.

Nuestro innovador concepto de telepráctica te dará la oportunidad de aprender mediante una experiencia inmersiva, que te proporcionará una integración más rápida y una visión mucho más realista de los contenidos: "Learning From an Expert"

Esta capacitación desarrollará los conceptos necesarios para trabajar en la creación de Interfaces, con la seguridad que el dominio de todas las áreas de conocimientos transversales a este tema, proporcionan al especialista. A lo largo del temario se plantearán abordajes didácticos innovadores, para conocer en profundidad la arquitectura de una aplicación distribuida. Se presentarán los aspectos particulares de la arquitectura cliente-servidor y se tratarán los fundamentos y desarrollos imprescindibles de los lenguajes de programación más utilizados, diferenciando entre lenguajes entre otras múltiples cuestiones fundamentales para el profesional.

Este conocimiento imprescindible se convierte además, en el primer paso para acceder a la propia capacidad de desarrollo de este tipo de tecnología.

A lo largo de esta capacitación se ofrece un panorama real de trabajo para poder valorar la conveniencia de su aplicación en el proyecto, valorando las indicaciones reales, la manera de desarrollo y las expectativas que se pueden tener con respecto a los resultados.

Mediante la experiencia, se consigue aprender cómo se desarrollan los conocimientos necesarios para avanzar en esta área de trabajo. Este aprendizaje, que necesita obligatoriamente de la experiencia, consigue en esta especialización, conciliar teleaprendizaje y enseñanza práctica ofreciendo una opción única para dar a su CV el impulso que busca.

Conviértete en uno de los profesionales más demandados del momento: capacitate en ingeniería informática a través del más completo y actualizado Máster Título Propio en Creación de Interfaces y Aplicaciones en Red"

Este **Máster Título Propio en Creación de Interfaces y Aplicaciones en Red** contiene el programa más completo y actualizado del mercado. Sus características más destacadas:

- Última tecnología en software de enseñanza online
- Sistema docente intensamente visual, apoyado en contenidos gráficos y esquemáticos de fácil asimilación y comprensión
- Desarrollo de casos prácticos presentados por expertos en activo
- Sistemas de vídeo interactivo de última generación
- Enseñanza apoyada en la telepráctica
- Sistemas de actualización y reciclaje permanente
- Aprendizaje autorregulable: total compatibilidad con otras ocupaciones
- Ejercicios prácticos de autoevaluación y constatación de aprendizaje
- Grupos de apoyo y sinergias educativas: preguntas al experto, foros de discusión y conocimiento
- Comunicación con el docente y trabajos de reflexión individual
- Disponibilidad de los contenidos desde cualquier dispositivo fijo o portátil con conexión a internet
- Bancos de documentación complementaria disponible permanentemente, incluso después del programa

Con un diseño metodológico que se apoya en técnicas de enseñanza contrastadas por su eficacia, este novedoso Máster Título Propio en Creación de Interfaces y Aplicaciones en Red te llevará a través de diferentes abordajes docentes para permitirte aprender de forma dinámica y eficaz"

El personal docente está integrado por profesionales de diferentes ámbitos relacionados con esta especialidad. De esta manera TECH Universidad FUNDEPOS se asegura de ofrecer el objetivo de actualización educativa que se pretende. Un cuadro multidisciplinar de profesionales capacitados y experimentados en diferentes entornos, que desarrollarán los conocimientos teóricos, de manera eficiente, pero, sobre todo, pondrán a el servicio los conocimientos prácticos derivados de su propia experiencia: una de las cualidades diferenciales de esta capacitación.

Este dominio de la materia se complementa con la eficacia del diseño metodológico. Elaborado por un equipo multidisciplinario de expertos en *e-learning* integra los últimos avances en tecnología educativa. De esta manera, podrá estudiar con un elenco de herramientas multimedia cómodas y versátiles que le darán la operatividad que necesita en su educación.

El diseño de este programa está basado en el Aprendizaje Basado en Problemas: un planteamiento que concibe el aprendizaje como un proceso eminentemente práctico. Para conseguirlo de forma remota, se usará la telepráctica: con la ayuda de un novedoso sistema de video interactivo, y el Learning From an Expert podrá adquirir los conocimientos como si estuviese enfrentándose al supuesto que está aprendiendo en ese momento. Un concepto que permitirá integrar y fijar el aprendizaje de una manera más realista y permanente.

Un Máster Título Propio que te capacitará para trabajar en todos los ámbitos de la Creación de Interfaces y Aplicaciones en Red con la solvencia de un profesional de alto nivel.

Con la experiencia de profesionales en activo que te llevarán a un conocimiento real, inmediato y concreto de este ámbito laboral.

02

Objetivos

El objetivo de TECH Universidad FUNDEPOS es otorgar información actualizada al profesional sobre la creación de interfaces y aplicaciones de red. Esto con el fin de que el ingeniero de sistemas adquiera nuevas destrezas para la realización de programas informáticos más completos. Una meta que, en tan apenas unos meses podrá ser adquirida a través de un plan de estudios enfocado en las necesidades actuales del sector.

Amplia tus conocimientos en computación e ingeniería del software y prepárate para competir entre los mejores del sector”

Objetivos generales

- Capacitar científica y tecnológicamente para el ejercicio de la ingeniería informática
- Obtener conocimientos amplios en el campo de la computación
- Obtener conocimientos amplios en el campo de la estructura de computadoras
- Adquirir los conocimientos necesarios en ingeniería del software
- Revisar las bases matemáticas, estadísticas y físicas imprescindibles para esta materia

Una vía de capacitación y crecimiento profesional que te impulsará hacia una mayor competitividad en el mercado laboral"

Objetivos específicos

Módulo 1. Interacción persona-ordenador

- Adquirir sólidos conocimientos relacionados con la interacción persona-ordenador y la creación de interfaces usables
- Entender la importancia de la usabilidad de las aplicaciones y el porqué hay que tenerlas en cuenta a la hora de diseñar nuestro software
- Comprender los distintos tipos de diversidad humanas, las limitaciones que suponen y cómo adaptar las interfaces de acuerdo a las necesidades específicas de cada una de ellas
- Aprender el proceso de diseño de interfaces, desde el análisis de requisitos hasta la evaluación, pasando por las distintas etapas intermedias necesarias para llevar realizar una interfaz adecuada
- Conocer las distintas pautas de accesibilidad, los estándares que las estableces y las herramientas que nos permiten evaluarla
- Entender los distintos métodos de interacción con el ordenador, mediante periféricos y dispositivos

Módulo 2. Bases de datos

- Aprender las distintas aplicaciones y propósitos de los sistemas de bases de datos, así como su funcionamiento y arquitectura
- Comprender el modelo relacional, desde su estructura y operaciones hasta el álgebra relacional extendida
- Aprender en profundidad qué son las bases de datos SQL, su funcionamiento, la definición de datos y la creación de consultas desde las más básicas hasta las más avanzadas y complejas

- Aprender a diseñar bases de datos usando el modelo entidad-relación, a crear diagramas y las características del modelo E-R extendido
- Profundizar en el diseño de bases de datos relacionales, analizando las distintas formas normales y los algoritmos de descomposición
- Sentar las bases para comprender el funcionamiento de las bases de datos NoSQL, así como introducir la base de datos Mongo DB

Módulo 3. Desarrollo de aplicaciones en red

- Conocer las características del lenguaje de marcado HTML y su uso en la creación web junto con las hojas de estilo CSS
- Aprender a utilizar el lenguaje de programación orientado al navegador JavaScript, y algunas de sus principales características
- Entender los conceptos de la programación orientada a componentes y de la arquitectura de componentes
- Aprender a usar el *Framework* para *Front-End Bootstrap* para el diseño de sitios web
- Entender la estructura del modelo vista controlador en el desarrollo de sitios web dinámicos
- Conocer la arquitectura orientada a servicios y las bases del protocolo HTTP

Módulo 4. Software libre y conocimiento abierto

- Aprender los conceptos de software libre y conocimiento abierto, así como los distintos tipos de licencias asociadas
- Conocer las principales herramientas libres disponibles en distintos ámbitos como sistemas operativos, gestión empresarial, gestores de contenido y creación de contenidos multimedia entre otras
- Entender la importancia y los beneficios del software libre en el mundo de la empresa, tanto por sus características como por sus costes
- Profundizar en el conocimiento del sistema operativo GNU/Linux, así como en las distintas distribuciones existentes, y cómo se pueden realizar adaptaciones personalizadas de ellas
- Aprender sobre el funcionamiento y desarrollo de WordPress, dado que este CMS supone más del 35% de las webs activas en el mundo, y más del 60% en el caso particular de los CMS
- Entender el funcionamiento del sistema operativo para dispositivos móviles Android, así como comprender las bases para el desarrollo de aplicaciones móviles tanto de forma nativa como con *Frameworks* multiplataforma

Módulo 5. Bases de datos avanzadas

- Introducir los distintos sistemas de bases de datos existentes actualmente en el mercado
- Aprender el uso de XML y de bases de datos para la web
- Comprender el funcionamiento de bases de datos avanzadas como son las bases de datos paralelas y las distribuidas
- Conocer la importancia de la indexación y la asociación en los sistemas de bases de datos
- Comprender el funcionamiento del procesamiento transaccional y los sistemas de recuperación
- Adquirir conocimientos relacionados con las bases de datos no relacionales y la minería de datos

Módulo 6. Ingeniería del software

- Conocer el marco de referencia de la ingeniería del software y la norma ISO/IEC 12207
- Aprender las características del proceso unificado de desarrollo de software y la planificación en el contexto del desarrollo de software ágil
- Conocer los distintos estilos de diseño de software distribuido y arquitecturas software orientadas a servicios

- Aprender los conceptos esenciales en el diseño de interfaces gráficas de usuario
- Comprender las bases del desarrollo de aplicaciones web
- Profundizar en las estrategias y técnicas de pruebas software, factores de calidad del software y distintas métricas utilizadas

Módulo 7. Programación avanzada

- Profundizar en los conocimientos de programación, especialmente en lo relaciona a la programación orientada a objetos, y los distintos tipos de relaciones entre clases existentes
- Conocer los distintos patrones de diseño para problemas orientados a objetos
- Aprender sobre la programación orientada a eventos y el desarrollo de interfaces de usuario con Qt
- Adquirir los conocimientos esenciales de la programación concurrente, los procesos y los hilos
- Aprender a gestionar el uso de los hilos y la sincronización, así como la resolución de los problemas comunes dentro de la programación concurrente
- Entender la importancia de la documentación y las pruebas en el desarrollo del software

Módulo 8. Reutilización del software

- Conocer el panorama general en la estrategia de la reutilización de software
- Aprender los distintos patrones relacionados con la reutilización de software, tanto de diseño, como de creación, estructurales y de comportamiento
- Introducir el concepto de *Framework*, así como conocer los principales tipos como los destinados al diseño de interfaces gráficas de usuario, al desarrollo de aplicaciones web y a la gestión de la persistencia de objetos en bases de datos
- Comprender el funcionamiento del patrón ampliamente usado actualmente de Modelo Vista Controlador (MVC)

Módulo 9. Inteligencia artificial e ingeniería del conocimiento

- Sentar las bases de la inteligencia artificial y la ingeniería del conocimiento, haciendo un breve recorrido por la historia de la inteligencia artificial hasta llegar a nuestros días
- Comprender los conceptos esenciales de la búsqueda en la inteligencia artificial, tanto de la búsqueda informada como de la no informada
- Entender el funcionamiento de la inteligencia artificial en juegos
- Aprender los conceptos fundamentales de las redes neuronales y el uso de los algoritmos genéticos
- Adquirir los mecanismos oportunos para representar el conocimiento, especialmente teniendo en cuenta la web semántica
- Comprender el funcionamiento de los sistemas expertos y los sistemas de soporte a la decisión

Módulo 10. Ingeniería del software avanzada

- Conocer en profundidad las distintas metodologías ágiles usadas en la ingeniería del software
- Aprender desarrollar usando las técnicas de Scrum, programación extrema y de desarrollo de software basado en reutilización
- Comprender los distintos patrones de arquitecturas de sistemas y de diseño de software, así como la arquitectura de las aplicaciones en la nube
- Aprender a realizar pruebas al software, con metodologías como *Test Driven Development, Acceptance Test Driven Development, Behavior Driven Development,* BDD y *Cucumber*
- Profundizar en la mejora del proceso de desarrollo de software y de calidad del software usando los estándares ISO/IEC
- Introducir el concepto de DevOps y cuáles son sus principales prácticas

03

Competencias

Este programa en Creación de Interfaces y Aplicaciones en Red ha sido creado como herramienta de alta capacitación para el profesional. Su intensivo plan de estudios contribuirá de manera significativa en el desarrollo y trabajo de programadores y portales web, aprovechando los recursos existentes en internet y los softwares libres, en una orientación práctica y útil.

Con las capacidades que este Máster Título Propio te permitirá adquirir podrás comenzar a desarrollar aplicaciones y trabajar como ingeniero de software, con la solvencia de una capacitación completa y actualizada"

Competencia general

- Adquirir las habilidades necesarias para el ejercicio profesional de la ingeniería informática con el conocimiento de todos los factores necesarios para realizarlo con calidad y solvencia

Una oportunidad creada para profesionales que buscan un programa intensivo y efectivo para dar un paso significativo en su profesión”

Competencias específicas

- Conocer en profundidad todas las facetas de la interacción persona-ordenaros y cómo involucran los desarrollos informáticos
- Ser solvente en el uso de bases de datos
- Desarrollar diferentes tipos de aplicaciones en red
- Describir y aprovechar el software libre y el conocimiento abierto existente en la red
- Trabajar como ingeniero de software
- Controlar el uso de las bases de datos avanzadas
- Realizar una programación avanzada
- Saber cómo se reutiliza el software
- Crear interfaces y aplicaciones en red
- Tener el dominio de los diferentes sistemas de trabajo en ingeniería de software avanzado

04

Estructura y contenido

Los contenidos de este Máster Título Propio han sido desarrollados por los diferentes expertos de este programa, con una finalidad clara: conseguir que el alumnado adquiera todas y cada una de las habilidades necesarias para convertirse en verdaderos expertos en esta materia.

Un programa completísimo y muy bien estructurado que le llevará hacia los más elevados estándares de calidad y éxito.

Un completísimo programa docente, estructurado en unidades didácticas completas y actualizadas en base a los últimos avances del sector, orientadas a un aprendizaje compatible con tu vida personal y profesional"

Módulo 1. Interacción persona-ordenador

1.1. Introducción a la interacción persona-ordenador
- 1.1.1. Qué es la interacción persona-ordenador
- 1.1.2. Relación de la interacción persona-ordenador con otras disciplinas
- 1.1.3. La interfaz de usuario
- 1.1.4. Usabilidad y accesibilidad
- 1.1.5. Experiencia de usuario y diseño centrado en el usuario

1.2. El ordenador y la interacción: interfaz de usuario y paradigmas de interacción
- 1.2.1. La interacción
- 1.2.2. Paradigmas y estilos de interacción
- 1.2.3. Evolución de las interfaces de usuario
- 1.2.4. Interfaces de usuario clásicas: WIMP/GUI, comandos, voz, realidad virtual
- 1.2.5. Interfaces de usuario innovadoras: móviles, portátiles, colaborativas, BCI

1.3. El factor humano: aspectos psicológicos y cognitivos
- 1.3.1. La importancia del factor humano en la interacción
- 1.3.2. El procesamiento humano de información
- 1.3.3. La entrada y salida de la información: visual, auditiva y táctil
- 1.3.4. Percepción y atención
- 1.3.5. Conocimiento y modelos mentales: representación, organización y adquisición

1.4. El factor humano: limitaciones sensoriales y físicas
- 1.4.1. Diversidad funcional, discapacidad y deficiencia
- 1.4.2. Diversidad visual
- 1.4.3. Diversidad auditiva
- 1.4.4. Diversidad cognitiva
- 1.4.5. Diversidad motórica
- 1.4.6. El caso de los inmigrantes digitales

1.5. El proceso de diseño (I): análisis de requisitos para el diseño de la interfaz de usuario
- 1.5.1. Diseño centrado en el usuario
- 1.5.2. Qué es el análisis de requisitos
- 1.5.3. La recogida de información
- 1.5.4. Análisis e interpretación de la información
- 1.5.5. Análisis de la usabilidad y la accesibilidad

1.6. El proceso de diseño (II): prototipado y análisis de tareas
- 1.6.1. Diseño conceptual
- 1.6.2. Prototipado
- 1.6.3. Análisis jerárquico de tareas

1.7. El proceso de diseño (III): la evaluación
- 1.7.1. Evaluación en el proceso de diseño: objetivos y métodos
- 1.7.2. Métodos de evaluación sin usuarios
- 1.7.3. Métodos de evaluación con usuarios
- 1.7.4. Estándares y normas de evaluación

1.8. Accesibilidad: definición y pautas
- 1.8.1. Accesibilidad y diseño universal
- 1.8.2. La iniciativa WAI y las pautas WCAG
- 1.8.3. Pautas WCAG 2.0 y 2.1

1.9. Accesibilidad: evaluación y diversidad funcional
- 1.9.1. Herramientas de evaluación de la accesibilidad en la web
- 1.9.2. Accesibilidad y diversidad funcional

1.10. El ordenador y la interacción: periféricos y dispositivos
- 1.10.1. Dispositivos y periféricos tradicionales
- 1.10.2. Dispositivos y periféricos alternativos
- 1.10.3. Móviles y tabletas
- 1.10.4. Diversidad funcional, interacción y periféricos

Módulo 2. Bases de datos

2.1. Aplicaciones y propósitos de los sistemas de base de datos
- 2.1.1. Aplicaciones de los diferentes sistemas de base de datos
- 2.1.2. Propósito en los diferentes sistemas de base de datos
- 2.1.3. Visión de los datos

2.2. Base de datos y arquitectura
- 2.2.1. Base de datos relacionales
- 2.2.2. El diseño de base de datos
- 2.2.3. Bases de datos basadas en objetos y semiestructuradas
- 2.2.4. Almacenamiento de datos y consultas
- 2.2.5. Gestión de transacciones
- 2.2.6. Minería y análisis de datos
- 2.2.7. Arquitectura de las bases de datos

2.3. El modelo relacional: estructura, operaciones y álgebra relacional extendida
- 2.3.1. La estructura de las BD relacionales
- 2.3.2. Operaciones fundamentales en el álgebra relacional
- 2.3.3. Otras operaciones del álgebra relacional
- 2.3.4. Operaciones del álgebra relacional extendida
- 2.3.5. Valores nulos
- 2.3.6. Modificación de la base de datos

2.4. SQL (I)
- 2.4.1. ¿Qué es SQL?
- 2.4.2. La definición de datos
- 2.4.3. Estructura básica de las consultas SQL
- 2.4.4. Operaciones sobre conjuntos
- 2.4.5. Funciones de agregación
- 2.4.6. Valores nulos

2.5. SQL (II)
- 2.5.1. Subconsultas anidadas
- 2.5.2. Consultas complejas
- 2.5.3. Vistas
- 2.5.4. Cursores
- 2.5.5. Consultas complejas
- 2.5.6. Disparadores

2.6. Diseño de base de datos y el modelo E-R
- 2.6.1. Visión general del proceso de diseño
- 2.6.2. El modelo entidad-relación
- 2.6.3. Restricciones

2.7. Diagramas entidad-relación
- 2.7.1. Diagramas entidad-relación
- 2.7.2. Aspectos del diseño entidad-relación
- 2.7.3. Conjuntos de entidades débiles

2.8. El modelo entidad-relación extendido
- 2.8.1. Características del modelo E-R extendido
- 2.8.2. Diseño de una base de datos
- 2.8.3. Reducción a esquemas relacionales

2.9. Diseño de bases de datos relacionales
- 2.9.1. Características de los buenos diseños relacionales
- 2.9.2. Dominios atómicos y la primera forma normal (1FN)
- 2.9.3. Descomposición mediante dependencias funcionales
- 2.9.4. Teoría de las dependencias funcionales
- 2.9.5. Algoritmos de descomposición
- 2.9.6. Descomposición mediante dependencias multivaloradas
- 2.9.7. Más formas normales
- 2.9.8. Proceso de diseño de las base de datos

2.10. Bases de datos NoSQL
- 2.10.1. ¿Qué son las bases de datos NoSQL?
- 2.10.2. Análisis de las diferentes opciones de NoSQL y sus características
- 2.10.3. Mongo DB

Módulo 3. Desarrollo de aplicaciones en red

3.1. Lenguajes de marcado HTML5
- 3.1.1. Conceptos básicos de HTML
- 3.1.2. Nuevos elementos HTML 5
- 3.1.3. Formularios: nuevos controles

3.2. Introducción a hojas de estilo CSS
- 3.2.1. Primeros pasos con CSS
- 3.2.2. Introducción a CSS3

3.3. Lenguaje script de navegador: JavaScript
- 3.3.1. Conceptos básicos de JavaScript
- 3.3.2. DOM
- 3.3.3. Eventos
- 3.3.4. JQuery
- 3.3.5. Ajax

3.4. Concepto de la programación orientada a componentes
- 3.4.1. Contexto
- 3.4.2. Componentes e interfaces
- 3.4.3. Estados de un componente

3.5. Arquitectura de componentes
- 3.5.1. Arquitecturas actuales
- 3.5.2. Integración y despliegue de componentes

3.6. *Framework Front-End: Bootstrap*
- 3.6.1. Diseño con rejilla
- 3.6.2. Formularios
- 3.6.3. Componentes

3.7. Modelo vista controlador
- 3.7.1. Métodos de desarrollo web
- 3.7.2. Patrón de diseño: MVC

3.8. Tecnologías *Grid* de la información
- 3.8.1. Incremento de recursos en computación
- 3.8.2. Concepto de tecnología *Grid*

3.9. Arquitectura orientada a servicios
- 3.9.1. SOA y servicios web
- 3.9.2. Topología de un servicio web
- 3.9.3. Plataformas para los servicios web

3.10. Protocolo HTTP
- 3.10.1. Mensajes
- 3.10.2. Sesiones persistentes
- 3.10.3. Sistema criptográfico
- 3.10.4. Funcionamiento del protocolo HTTPS

Módulo 4. Software libre y conocimiento abierto

4.1. Introducción al software libre
- 4.1.1. Historia del software libre
- 4.1.2. "Libertad" en el software
- 4.1.3. Licencias de uso de herramientas software
- 4.1.4. Propiedad intelectual del software
- 4.1.5. ¿Cuál es la motivación de usar software libre?
- 4.1.6. Mitos del software libre
- 4.1.7. Top500

4.2. Conocimiento abierto y licencias CC
- 4.2.1. Conceptos básicos
- 4.2.2. Licencias *Creative Commons*
- 4.2.3. Otras licencias de contenidos
- 4.2.4. Wikipedia y otros proyectos de conocimiento abierto

4.3. Principales herramientas de software libre
- 4.3.1. Sistemas operativos
- 4.3.2. Aplicaciones ofimáticas
- 4.3.3. Aplicaciones de gestión empresarial
- 4.3.4. Gestores de contenido web
- 4.3.5. Herramientas de creación de contenidos multimedia
- 4.3.6. Otras aplicaciones

4.4. La empresa: el software libre y sus costes
- 4.4.1. Software libre: ¿sí o no?
- 4.4.2. Verdades y mentiras sobre el software libre
- 4.4.3. Software empresarial basado en software libre
- 4.4.4. Costes del software
- 4.4.5. Modelos de software libre

4.5. El sistema operativo GNU/Linux
- 4.5.1. Arquitectura
- 4.5.2. Estructura de directorios básica
- 4.5.3. Características y estructura del sistema de archivos
- 4.5.4. Representación interna de los archivos

4.6. El sistema operativo móvil Android
- 4.6.1. Historia
- 4.6.2. Arquitectura
- 4.6.3. Forks de Android
- 4.6.4. Introducción al desarrollo para Android
- 4.6.5. *Frameworks* para el desarrollo de aplicaciones móviles

4.7. Creación de sitios web con WordPress
- 4.7.1. Características y estructura de WordPress
- 4.7.2. Creación de sitios en wordpress.com
- 4.7.3. Instalación y configuración de WodPress en un servidor propio
- 4.7.4. Instalación de plugins y ampliación de WordPress
- 4.7.5. Creación de plugins para WordPress
- 4.7.6. Creación de temas para WordPress

4.8. Las tendencias del software libre
- 4.8.1. Entrornos en la nube
- 4.8.2. Herramientas de monitorización
- 4.8.3. Sistemas operativos
- 4.8.4. Big Data y Open Data 2.0
- 4.8.5. Computación cuántica

4.9. Control de versiones
- 4.9.1. Conceptos básicos
- 4.9.2. Git
- 4.9.3. Servicios Git en la nube y autoalojados
- 4.9.4. Otros sistemas de control de versiones

4.10. Distribuciones de GNU/Linux personalizadas
- 4.10.1. Principales distribuciones
- 4.10.2. Distribuciones derivadas de Debian
- 4.10.3. Creación de paquetes deb
- 4.10.4. Modificación de la distribución
- 4.10.5. Generación de imágenes ISO

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

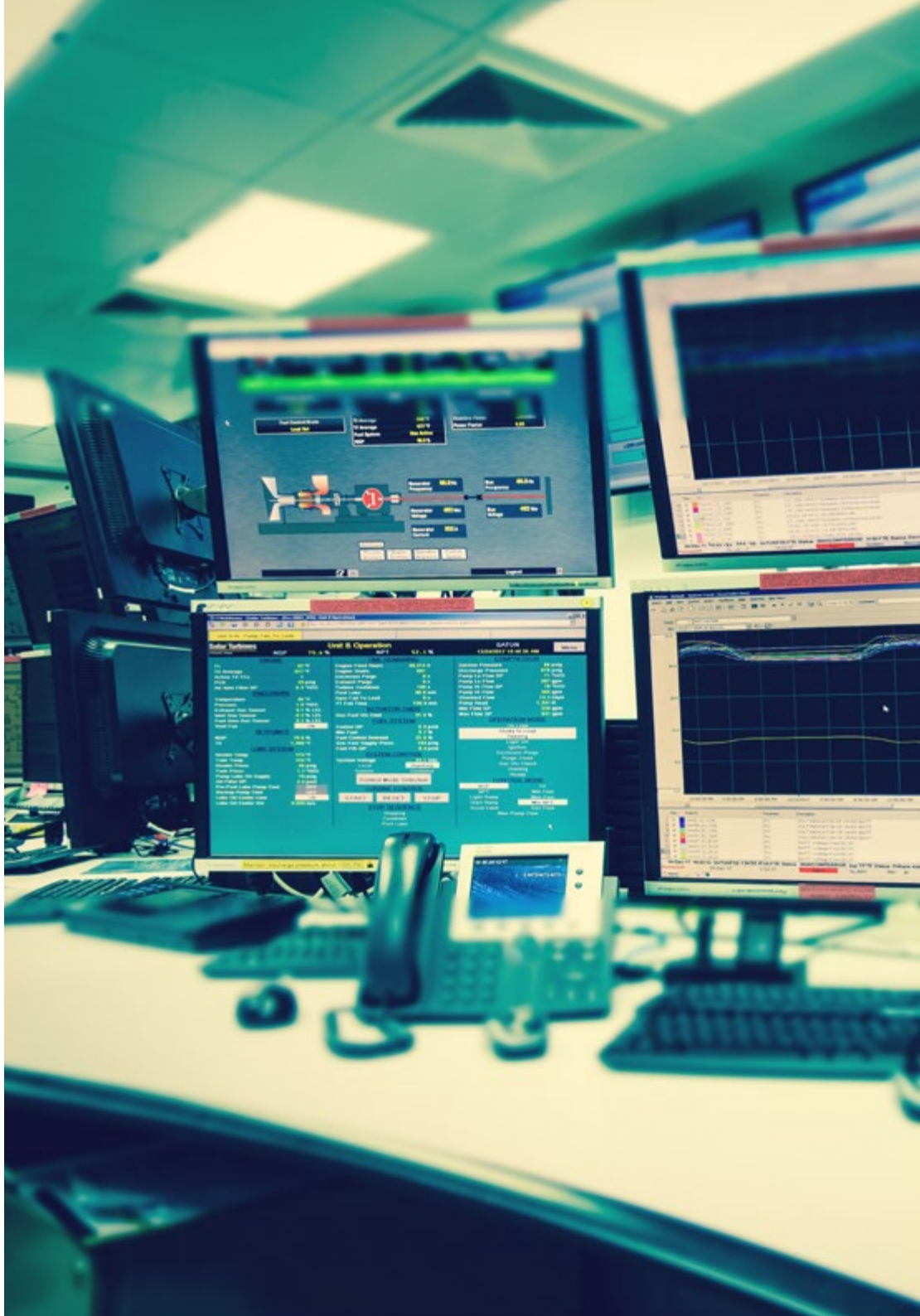
5.2. XML y bases de datos para la web
- 5.2.1. Validación de documentos XML
- 5.2.2. Transformaciones de documentos XML
- 5.2.3. Almacenamiento de datos XML
- 5.2.4. Bases de datos relacionales XML
- 5.2.5. SQL/XML
- 5.2.6. Bases de datos nativas XML

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. Ingeniería del software

6.1. Marco de Ingeniería software
- 6.1.1. Características del software
- 6.1.2. Los procesos principales en Ingeniería del software
- 6.1.3. Modelos de proceso de desarrollo software
- 6.1.4. Marco de referencia estándar para el proceso de desarrollo de software: la norma ISO/IEC 12207

6.2. Proceso unificado de desarrollo software
- 6.2.1. Proceso unificado
- 6.2.2. Dimensiones del proceso unificado
- 6.2.3. Proceso de desarrollo dirigido por casos de uso
- 6.2.4. Flujos de trabajo fundamentales de procesos unificados

6.3. Planificación en el contexto de desarrollo de software ágil
- 6.3.1. Características del desarrollo software ágil
- 6.3.2. Diferentes horizontes temporales de planificación en el desarrollo ágil
- 6.3.3. Marco de desarrollo ágil Scrum y horizontes temporales de planificación
- 6.3.4. Historias de usuario como unidad de planificación y estimación
- 6.3.5. Técnicas comunes para derivar una estimación
- 6.3.6. Escalas para interpretar las estimaciones
- 6.3.7. *Planning poker*
- 6.3.8. Tipos de planificaciones comunes: planificación de entregas y planificación de iteración

6.4. Estilos de diseño de software distribuido y arquitecturas software orientadas a servicios
- 6.4.1. Modelos de comunicación en sistemas software distribuidos
- 6.4.2. Capa intermedia o middleware
- 6.4.3. Patrones de arquitectura para sistemas distribuidos
- 6.4.4. Proceso general de diseño de servicios software
- 6.4.5. Aspectos de diseño de servicios software
- 6.4.6. Composición de servicios
- 6.4.7. Arquitectura de servicios web
- 6.4.8. Componentes de Infraestructura y SOA

6.5. Introducción al desarrollo software dirigido por modelos
- 6.5.1. El concepto de modelo
- 6.5.2. Desarrollo software dirigido por modelos
- 6.5.3. Marco de referencia de desarrollo dirigido por modelos MDA
- 6.5.4. Elementos de un modelo de transformación

6.6. Diseño de interfaces gráficas de usuario
- 6.6.1. Principios de diseño de interfaces de usuario
- 6.6.2. Patrones de diseño arquitectónico para sistemas interactivos: Modelo Vista Controlador (MVC)
- 6.6.3. Experiencia de usuario (UX *User Experience*)
- 6.6.4. Diseño centrado en el usuario
- 6.6.5. Proceso de análisis y diseño de la interfaz gráfica de usuario
- 6.6.6. Usabilidad de interfaces de usuario
- 6.6.7. Accesibilidad en interfaces de usuario

6.7. Diseño de aplicaciones web
- 6.7.1. Características de las aplicaciones web
- 6.7.2. Interfaz de usuario de una aplicación web
- 6.7.3. Diseño de navegación
- 6.7.4. Protocolo de interacción base para aplicaciones web
- 6.7.5. Estilos de arquitectura para aplicaciones web

6.8. Estrategias y técnicas de pruebas software y factores de calidad del software
- 6.8.1. Estrategias de prueba
- 6.8.2. Diseños de casos de prueba
- 6.8.3. Relación coste calidad
- 6.8.4. Modelos de calidad
- 6.8.5. Familia de normas ISO/IEC 25000 (SQuaRE)
- 6.8.6. Modelo de calidad de producto (ISO 2501n)
- 6.8.7. Modelos de calidad de datos (ISO 2501n)
- 6.8.8. Gestión de la calidad del software

6.9. Introducción a las métricas en ingeniería software
- 6.9.1. Conceptos básicos: medidas, métricas e indicadores
- 6.9.2. Tipos de métricas en ingeniería software
- 6.9.3. El proceso de medición
- 6.9.4. ISO 25024. Métricas externas y de calidad en uso
- 6.9.5. Métrica orientada a objetos

6.10. Mantenimiento y reingeniería software
- 6.10.1. Proceso de mantenimiento
- 6.10.2. Marco estándar de proceso de mantenimiento. ISO/EIEC 14764
- 6.10.3. Modelo de proceso de reingeniería de software
- 6.10.4. Ingeniería inversa

Módulo 7. Programación avanzada

7.1. Introducción a la programación orientada a objetos
- 7.1.1. Introducción a la programación orientada a objetos
- 7.1.2. Diseño de clases
- 7.1.3. Introducción a UML para el modelado de los problemas

7.2. Relaciones entre clases
- 7.2.1. Abstracción y herencia
- 7.2.2. Conceptos avanzados de herencia
- 7.2.3. Polimorfismo
- 7.2.4. Composición y agregación

7.3. Introducción a los patrones de diseño para problemas orientados a objetos
- 7.3.1. ¿Qué son los patrones de diseño?
- 7.3.2. Patrón *Factory*
- 7.3.3. Patrón Singleton
- 7.3.4. Patrón *Observer*
- 7.3.5. Patrón *Composite*

7.4. Excepciones
- 7.4.1. ¿Qué son las excepciones?
- 7.4.2. Captura y gestión de excepciones
- 7.4.3. Lanzamiento de excepciones
- 7.4.4. Creación de excepciones

7.5. Interfaces de usuarios
- 7.5.1. Introducción a Qt
- 7.5.2. Posicionamiento
- 7.5.3. ¿Qué son los eventos?
- 7.5.4. Eventos: definición y captura
- 7.5.5. Desarrollo de interfaces de usuario

7.6. Introducción a la programación concurrente
- 7.6.1. Introducción a la programación concurrente
- 7.6.2. El concepto de proceso e hilo
- 7.6.3. Interacción entre procesos o hilos
- 7.6.4. Los hilos en C++
- 7.6.5. Ventajas e inconvenientes de la programación concurrente

7.7. Gestión de hilos y sincronización
- 7.7.1. Ciclo de vida de un hilo
- 7.7.2. La clase Thread
- 7.7.3. Planificación de hilos
- 7.7.4. Grupos hilos
- 7.7.5. Hilos de tipo demonio
- 7.7.6. Sincronización
- 7.7.7. Mecanismos de bloqueo
- 7.7.8. Mecanismos de comunicación
- 7.7.9. Monitores

7.8. Problemas comunes dentro de la programación concurrente
- 7.8.1. El problema de los productores consumidores
- 7.8.2. El problema de los lectores y escritores
- 7.8.3. El problema de la cena de los filósofos

7.9. Documentación y pruebas de software
- 7.9.1. ¿Por qué es importante documentar el software?
- 7.9.2. Documentación de diseño
- 7.9.3. Uso de herramientas para la documentación

7.10. Pruebas de software
- 7.10.1. Introducción a las pruebas del software
- 7.10.2. Tipos de pruebas
- 7.10.3. Prueba de unidad
- 7.10.4. Prueba de integración
- 7.10.5. Prueba de validación
- 7.10.6. Prueba del sistema

Módulo 8. Reutilización del software

8.1. Panorama general de la reutilización de software
- 8.1.1. En qué consiste la reutilización del software
- 8.1.2. Ventajas e inconvenientes de la reutilización de software
- 8.1.3. Principales técnicas de reutilización de software

8.2. Introducción a los patrones de diseño
- 8.2.1. ¿Qué es un patrón de diseño?
- 8.2.2. Catálogo de los principales patrones de diseño
- 8.2.3. Cómo usar patrones para resolver problemas de diseño
- 8.2.4. Cómo seleccionar el mejor patrón de diseño

8.3. Patrones de creación
- 8.3.1. Patrones de creación
- 8.3.2. Patrón *Abstract Factory*
- 8.3.3. Ejemplo de implementación del patrón *Abstract Factory*
- 8.3.4. Patrón Builder
- 8.3.5. Ejemplo de implementación del Builder
- 8.3.6. Patrón Abstract Factory vs. Builder

8.4. Patrones de creación (II)
- 8.4.1. Patrón Factory Method
- 8.4.2. *Factory Method* vs. *Abstract Factory*
- 8.4.3. Patrón Singleton

8.5. Patrones estructurales
- 8.5.1. Patrones estructurales
- 8.5.2. Patrón *Adapter*
- 8.5.3. Patrón *Bridge*

8.6. Patrones estructurales (II)
- 8.6.1. Patrón *Composite*
- 8.6.2. Patrón decorador

8.7. Patrones estructurales (III)
- 8.7.1. Patrón Facade
- 8.7.2. Patrón Proxy

8.8. Patrones de comportamiento
- 8.8.1. Concepto de los patrones de comportamiento
- 8.8.2. Patrón de comportamiento: cadena de responsabilidad
- 8.8.3. Patrón de comportamiento orden

8.9. Patrones de comportamiento (II)
- 8.9.1. Patrón intérprete o *Interpreter*
- 8.9.2. Patrón iterador
- 8.9.3. Patrón observador
- 8.9.4. Patrón estrategia

8.10. *Frameworks*
- 8.10.1. Concepto de *Frameworks*
- 8.10.2. Desarrollo mediante *Frameworks*
- 8.10.3. Patrón *Model View Controller*
- 8.10.4. *Framework* para diseño de interfaces gráficas de usuario
- 8.10.5. *Frameworks* para el desarrollo de aplicaciones web
- 8.10.6. *Frameworks* para la gestión de la persistencia de objetos en bases de datos

Módulo 9. Inteligencia artificial e ingeniería del conocimiento

9.1. Introducción a la inteligencia artificial y a la ingeniería del conocimiento
- 9.1.1. Breve historia de la inteligencia artificial
- 9.1.2. La inteligencia artificial hoy en día
- 9.1.3. Ingeniería del conocimiento

9.2. Búsqueda
- 9.2.1. Conceptos comunes de búsqueda
- 9.2.2. Búsqueda no informada
- 9.2.3. Búsqueda informada

9.3. Satisfacibilidad booleana, satisfacibilidad de restricciones y planificación automática
- 9.3.1. Satisfacibilidad booleana
- 9.3.2. Problemas de satisfacción de restricciones
- 9.3.3. Planificación automática y PDDL
- 9.3.4. Planificación como búsqueda heurística
- 9.3.5. Planificación con SAT

9.4. La inteligencia artificial en juegos
- 9.4.1. Teoría de juegos
- 9.4.2. Minimax y poda Alfa-Beta
- 9.4.3. Simulación: Monte Carlo

9.5. Aprendizaje supervisado y no supervisado
- 9.5.1. Introducción al aprendizaje automático
- 9.5.2. Clasificación
- 9.5.3. Regresión
- 9.5.4. Validación de resultados
- 9.5.5. Agrupación (*Clustering*)

9.6. Redes de neuronas
- 9.6.1. Fundamentos biológicos
- 9.6.2. Modelo computacional
- 9.6.3. Redes de neuronas supervisadas y no supervisadas
- 9.6.4. Perceptrón simple
- 9.6.5. Perceptrón multicapa

9.7. Algoritmos genéticos
- 9.7.1. Historia
- 9.7.2. Base biológica
- 9.7.3. Codificación de problemas
- 9.7.4. Generación de la población inicial
- 9.7.5. Algoritmo principal y operadores genéticos
- 9.7.6. Evaluación de individuos: fitness

9.8. Tesauros, vocabularios, taxonomías
- 9.8.1. Vocabularios
- 9.8.2. Taxonomías
- 9.8.3. Tesauros
- 9.8.4. Ontologías

9.9. Representación del conocimiento: web semántica
- 9.9.1. Web semántica
- 9.9.2. Especificaciones: RDF, RDFS y OWL
- 9.9.3. Inferencia/razonamiento
- 9.9.4. *Linked* Data

9.10. Sistemas expertos y DSS
- 9.10.1. Sistemas expertos
- 9.10.2. Sistemas de soporte a la decisión

Módulo 10. Ingeniería del software avanzada

10.1. Introducción a las metodologías ágiles
- 10.1.1. Modelos de proceso y metodologías
- 10.1.2. Agilidad y procesos ágiles
- 10.1.3. Manifiesto ágil
- 10.1.4. Algunas metodologías ágiles
- 10.1.5. Ágil vs. Tradicional

10.2. Scrum
- 10.2.1. Orígenes y filosofía de Scrum
- 10.2.2. Valores de Scrum
- 10.2.3. Flujo del proceso Scrum
- 10.2.4. Los roles de Scrum
- 10.2.5. Los artefactos de Scrum
- 10.2.6. Los eventos de Scrum
- 10.2.7. Las historias de usuario
- 10.2.8. Extensiones de Scrum
- 10.2.9. Estimaciones ágiles
- 10.2.10. Escalado de Scrum

10.3. Programación extrema
- 10.3.1. Justificación y visión general de XP
- 10.3.2. El ciclo de vida en XP
- 10.3.3. Los cinco valores básicos
- 10.3.4. Las doce prácticas básicas en XP
- 10.3.5. Roles de los participantes
- 10.3.6. XP industrial
- 10.3.7. Valoración crítica de XP

10.4. Desarrollo de software basado en reutilización
- 10.4.1. La reutilización del software
- 10.4.2. Niveles de reutilización de código
- 10.4.3. Técnicas concretas de reutilización
- 10.4.4. Desarrollo basado en componentes
- 10.4.5. Beneficios y problemas de la reutilización
- 10.4.6. Planificación de la reutilización

10.5. Patrones de arquitectura de sistemas y de diseño de software
- 10.5.1. El diseño arquitectónico
- 10.5.2. Patrones arquitectónicos generales
- 10.5.3. Arquitecturas tolerantes a fallos
- 10.5.4. Arquitecturas de sistemas distribuidos
- 10.5.5. Los patrones de diseño
- 10.5.6. Patrones de Gamma
- 10.5.7. Patrones de diseño de interacción

10.6. Arquitectura de aplicaciones en la nube
- 10.6.1. Fundamentos de *Cloud Computing*
- 10.6.2. Calidad de las aplicaciones en la nube
- 10.6.3. Estilos de arquitectura
- 10.6.4. Patrones de diseño

10.7. Pruebas del software: TDD, ATDD y BDD
- 10.7.1. Verificación y validación del software
- 10.7.2. Las pruebas de software
- 10.7.3. *Test Driven Development* (TDD)
- 10.7.4. *Acceptance Test Driven Development* (ATDD)
- 10.7.5. *Behavior Driven Development* (BDD)
- 10.7.6. BDD y *Cucumber*

10.8. La mejora del proceso de software
- 10.8.1. La mejora del proceso de software
- 10.8.2. El proceso de mejora de procesos
- 10.8.3. Modelos de madurez
- 10.8.4. El modelo CMMI
- 10.8.5. CMMI V2.0
- 10.8.6. CMMI y ágil

10.9. La calidad del producto software: SQuaRE
- 10.9.1. La calidad del software
- 10.9.2. Modelos de calidad del producto software
- 10.9.3. Familia ISO/IEC 25000
- 10.9.4. ISO/IEC 25010: modelo y características de calidad
- 10.9.5. ISO/IEC 25012: la calidad de los datos
- 10.9.6. ISO/IEC 25020: medición de la calidad del software
- 10.9.7. ISO/IEC 25022, 25023 y 25024: métricas de calidad del software y de los datos
- 10.9.8. ISO/IEC 25040: evaluación del software
- 10.9.9. El proceso de certificación

10.10. Introducción a DevOps
- 10.10.1. Concepto de DevOps
- 10.10.2. Prácticas principales

Una experiencia única, clave y decisiva para impulsar tu desarrollo profesional"

0?

Metodología

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

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

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

Estudio de Caso para contextualizar todo el contenido

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

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

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

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

Un método de aprendizaje innovador y diferente

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

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

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

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

Relearning Methodology

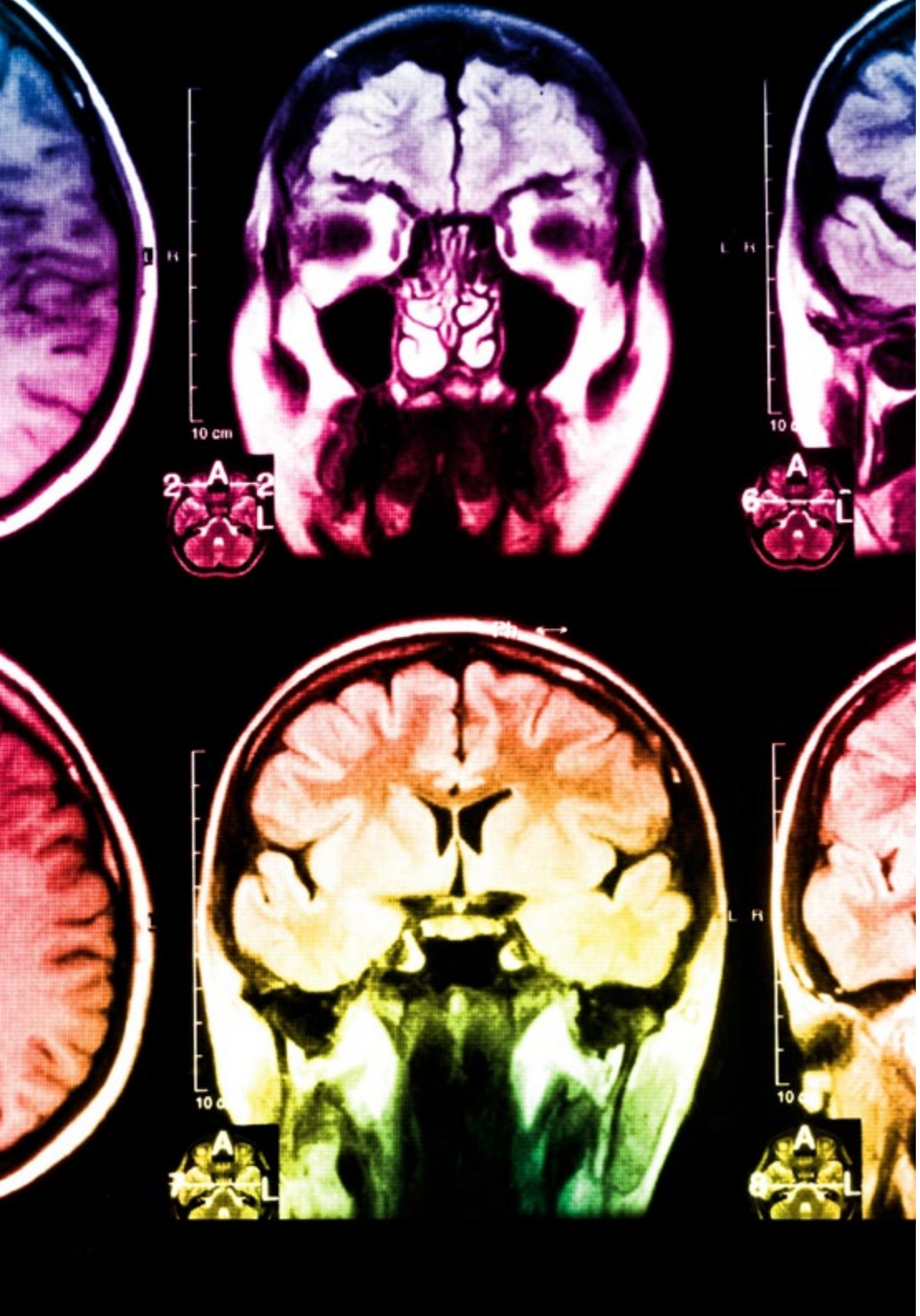
TECH Universidad FUNDEPOS aúna de forma eficaz la metodología del Estudio de Caso con un sistema de aprendizaje 100% online basado en la reiteración, que combina elementos didácticos diferentes en cada lección.

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

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

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

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



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

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

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

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

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

Material de estudio

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

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

Clases magistrales

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

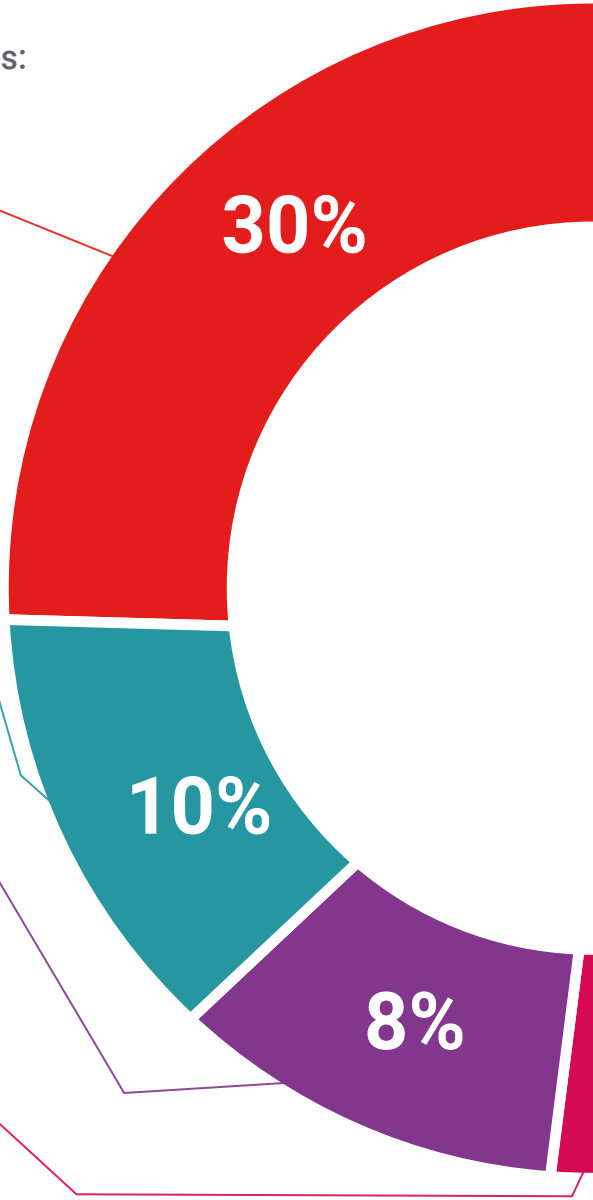
El denominado Learning from an Expert afianza el conocimiento y el recuerdo, y genera seguridad en las futuras decisiones difíciles.

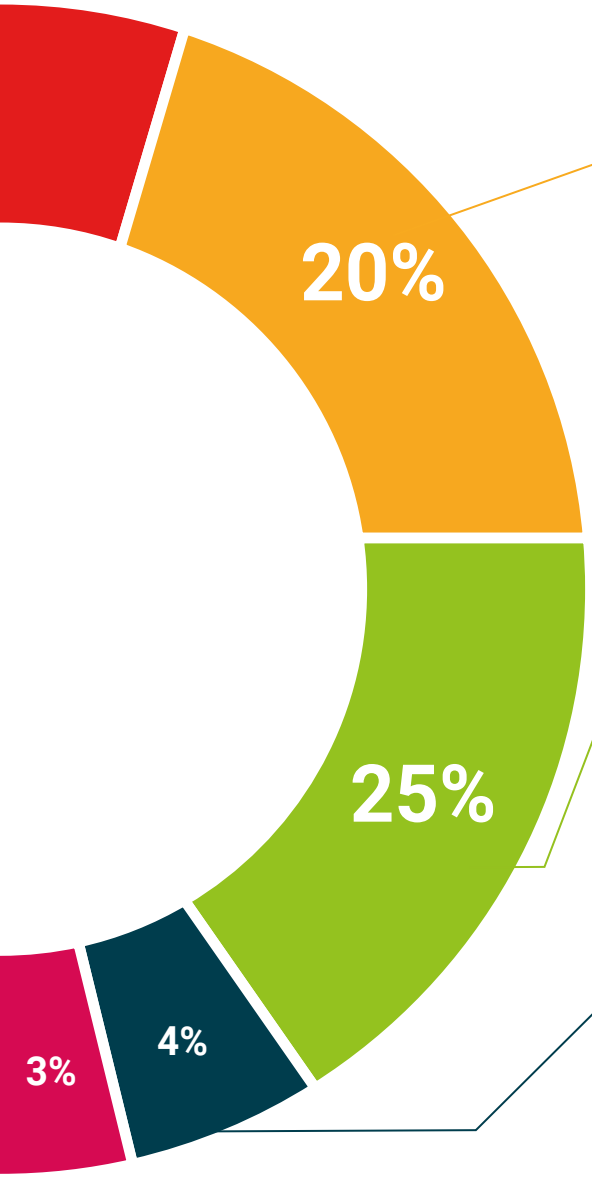
Prácticas de habilidades y competencias

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

Lecturas complementarias

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





Case studies

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

Resúmenes interactivos

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

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

Testing & Retesting

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

06

Titulación

El Máster Título Propio en Creación de Interfaces y Aplicaciones en Red garantiza, además de la capacitación más rigurosa y actualizada, el acceso a dos diplomas de Máster Propio, uno expedido por TECH Universidad Tecnológica 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 Creación de Interfaces y Aplicaciones en Red** es el más completo del panorama académico actual. A su egreso, el estudiante recibirá un diploma universitario emitido por TECH Universidad Tecnológica, y otro por Universidad FUNDEPOS.

Estos títulos de formación permanente y actualización profesional de TECH Universidad Tecnológica 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 Creación de Interfaces y Aplicaciones en Red**

N.º Horas: **1.500 h.**

tech universidad tecnológica

Otorga la presente

CONSTANCIA

a

C. _____, con documento de identificación nº_____

Por haber superado con éxito y acreditado el programa de

MÁSTER TÍTULO PROPIO

en

Creación de Interfaces y Aplicaciones en Red

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

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

A 17 de junio de 2020

Mtra.Tere Guevara Navarro
Rectora

código único TECH: AFWOR23S techtitute.com/titulos

UNIVERSIDAD FUNDEPOS DE COSTA RICA

Confiere el presente certificado de aprovechamiento a:

NOMBRE COMPLETO

Por aprobar satisfactoriamente el programa:

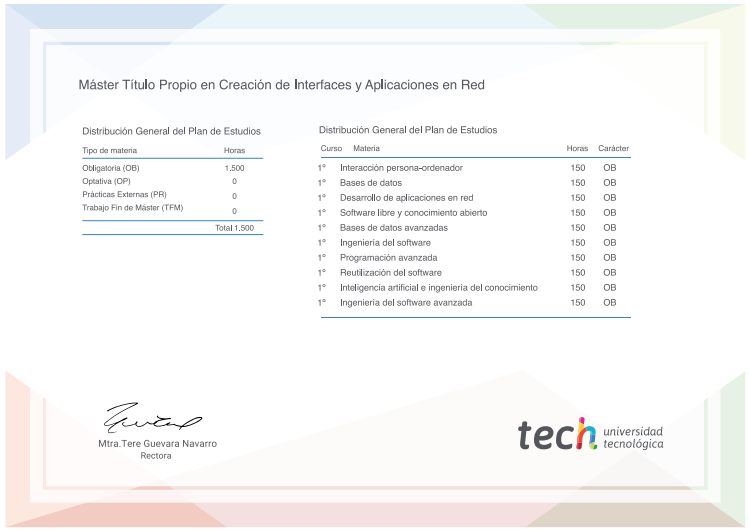
CREACIÓN DE INTERFACES Y APLICACIONES EN RED

Título Propio. Duración: 1500 horas.
Impartido del 15/12/2023 y fecha de finalización 14/12/2024.

CERTIFICADO DE CALIDAD 2023

Rubén Mora
DECANO

Este título propio se deberá acompañar siempre del título universitario habilitan expedido por la autoridad competente para ejercer profesionalmente en cada país. Código.


Máster Título Propio en Creación de Interfaces y Aplicaciones en Red

Distribución General del Plan de Estudios

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

Distribución General del Plan de Estudios

Curso	Materia	Horas	Carácter
1º	Interacción persona-ordenador	150	OB
1º	Bases de datos	150	OB
1º	Desarrollo de aplicaciones en red	150	OB
1º	Software libre y conocimiento abierto	150	OB
1º	Bases de datos avanzadas	150	OB
1º	Ingeniería del software	150	OB
1º	Programación avanzada	150	OB
1º	Reutilización del software	150	OB
1º	Inteligencia artificial e ingeniería del conocimiento	150	OB
1º	Ingeniería del software avanzada	150	OB

Mtra.Tere Guevara Navarro
Rectora

tech universidad tecnológica

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

tech universidad FUNDEPOS

Máster Título Propio
Creación de Interfaces y Aplicaciones en Red

- Modalidad: **online**
- Duración: **12 meses**
- Titulación: **TECH Universidad FUNDEPOS**
- Dedicación: **16h/semana**
- Horario: **a tu ritmo**
- Exámenes: **online**

Máster Título Propio

Creación de Interfaces y Aplicaciones en Red

tech universidad FUNDEPOS