CURSO DE PROGRAMACIÓN PARA GESTIÓN DE SISTEMAS

ESCUELA SUPERIOR DE ESTUDIOS PROFESIONALES

GRUPO EDUCATIVO ESBA

ESEP ESCUELA SUPERIOR DE ESTUDIOS PROFESIONALES

ESBA ESTUDIOS SUPERIORES DE BUENOS AIRES

“Educación de calidad al alcance de todos”

- ✓ Más de 75.000 egresados
- ✓ 40 años de experiencia

• DESCRIPCIÓN DE LA FORMACIÓN •

Este Curso de Programación para Gestión de Sistemas le ofrece una formación especializada en la materia. Debemos saber que en la actualidad, en el mundo de la informática y las comunicaciones, es muy importante conocer la programación de sistemas informáticos, dentro del área profesional de sistemas y telemática. Por ello, con el presente curso se trata de aportar los conocimientos necesarios para conocer el desarrollo y optimización de componentes software para tareas administrativas de sistemas y el desarrollo de componentes software para el manejo de dispositivos (Drivers).

• OBJETIVOS •

- Analizar las especificaciones técnicas proporcionadas para el desarrollo a realizar, comprobando su validez y optimización
- Desarrollar elementos software destinados a la gestión de los recursos del sistema, mediante herramientas y lenguajes de programación de sistemas
- Utilizar las técnicas y estándares utilizadas en el desarrollo, distribución e implantación de manejadores de dispositivos (drivers), para la integración de periféricos en el sistema informático según especificaciones técnicas y funcionales dadas

• DESTINATARIOS •

Este curso está dirigido a todas aquellas personas que se dedican al mundo de la informática y las comunicaciones, concretamente en programación de sistemas informáticos, dentro del área profesional de sistemas y telemática y a todas aquellas personas interesadas en adquirir conocimientos relacionados con la Programación para Gestión de Sistemas.

• PREPARA AL ALUMNO PARA •

Adquirir los conocimientos necesarios para conocer el desarrollo y optimización de componentes software para tareas administrativas de sistemas y el desarrollo de componentes software para el manejo de dispositivos (Drivers).

• TITULACIÓN •

Al finalizar el Curso de Auxiliar de Guardería obtendrás una titulación emitida por la Escuela Superior de Estudios Profesionales y el grupo educativo ESBA.

DURACIÓN: 180hs

• ¿QUÉ DISTINGUE A ESTE CURSO DE LOS DEMÁS? •

La Escuela Superior de Estudios Profesionales, en conjunto con Grupo Educativo ESBA, unen su experiencia y trayectoria para ofrecerte esta capacitación de una manera clara, didáctica y sencilla, que te permitirá obtener importantes conocimientos en la temática, en clases virtuales, sin días ni horarios fijos de cursada y a tu propio ritmo.

• TEMARIO •

UNIDAD FORMATIVA 1. DESARROLLO Y OPTIMIZACIÓN DE COMPONENTES SOFTWARE PARA TAREAS ADMINISTRATIVAS DE SISTEMAS

UNIDAD DIDÁCTICA 1. DESCRIPCIÓN DE LOS SERVICIOS, ESTRUCTURA Y ADMINISTRACIÓN DE SISTEMAS OPERATIVOS.

1. Definición y conceptos básicos sobre Sistemas Operativos

2. Descripción de los servicios básicos ofrecidos por un Sistema Operativo
3. Gestión de memoria. Memoria virtual
4. Ejecución de programas y gestión de procesos
5. Gestión del almacenamiento. Sistemas de Archivos
6. Gestión de dispositivos de entrada/salida
7. Gestión de red
8. Gestión de errores
9. Gestión de la seguridad
10. Auditoría (logs del sistema)
11. Procesos de arranque (boot) y finalización del sistema (shutdown)
12. Características estructurales de los Sistemas Operativos:
13. Sistemas monolíticos
14. Microkernels
15. Sistemas modulares y por capas
16. Máquinas virtuales
17. Sistemas distribuidos
18. Herramientas administrativas de uso común en Sistemas Operativos:
19. Interfaces de usuario gráficos
20. Intérpretes de comandos

UNIDAD DIDÁCTICA 2. PROGRAMACIÓN DE SISTEMAS OPERATIVOS. LENGUAJES Y LIBRERÍAS DE USO COMÚN.

1. Las llamadas al sistema (System Calls):
2. Definición
3. Uso directo y mediante Application Programming Interfaces (APIs)
4. Principales tipos de llamadas al sistema:
5. Control de procesos
6. Gestión de ficheros
7. Gestión de dispositivos
8. Información del sistema
9. Comunicaciones
10. Descripción y uso de las APIs estándar de uso común para llamadas a sistema:
11. Win32 API (Sistemas Windows)
12. POSIX API (Sistemas Unix, Linux, Mac)
13. Java API (Multiplataforma)

14. Programas de utilidades y comandos del sistema:
15. Principales tipos:
16. Operaciones con ficheros y directorios
17. Funciones de estado
18. Edición y manipulación de ficheros
19. Soporte para lenguajes de programación (compiladores, enlazadores, ensambladores, intérpretes, etc.)
20. Ejecución de programas
21. Comunicaciones, mensajería, intercambio remoto de archivos, etc.
22. Uso de utilidades y comandos mediante lenguajes de script de uso común
23. Windows scripting
24. Linux/Unix scripting

UNIDAD DIDÁCTICA 3. EL CICLO DE VIDA DEL SOFTWARE DE GESTIÓN DE SISTEMAS.

1. Modelos del ciclo de vida del software.
2. En cascada (waterfall)
3. Iterativo
4. Incremental
5. En V
6. Basado en componentes (CBSE)
7. Desarrollo rápido (RAD)
8. Ventajas e inconvenientes. Pautas para la selección de la metodología más adecuada.
9. Descripción de las fases en el ciclo de vida del software:
10. Análisis y especificación de requisitos
11. Tipos de requisitos : funcionales/ no funcionales, de usuario, de interfaz, de seguridad y de rendimiento
12. Modelos para el análisis de requisitos
13. Documentación de requisitos
14. Validación de requisitos
15. Gestión de requisitos
16. Diseño:
17. Modelos para el diseño de sistemas : contexto y arquitectura, procesos, datos, objetos, interfaces de usuario ,componentes y despliegues

18. Diagramas de diseño: diagramas de entidad-relación, diagramas de flujo, diagramas de contexto y UML. Diagramas UML de uso común en diseño de sistemas
19. Documentación: herramientas de generación de documentación y documentación el código
20. Implementación. Conceptos generales de desarrollo de software:
21. Principios básicos del desarrollo de software
22. Técnicas de desarrollo de software : basadas en prototipos, basadas en componentes, métodos de desarrollo rápido y otras técnicas de desarrollo
23. Validación, verificación y pruebas:
24. Validación y verificación de sistemas: planificación, métodos formales de verificación y métodos automatizados de análisis
25. Pruebas de software: tipos, diseño de pruebas, ámbito de aplicación, automatización de pruebas, herramientas y estándares sobre pruebas de software.
26. Calidad del software:
27. Principios de calidad del software
28. Métricas y calidad del software:
29. Concepto de métrica y su importancia en la medición de la calidad
30. Principales métricas en las fases del ciclo de vida software
31. Estándares para la descripción de los factores de Calidad:
32. ISO-9126
33. Otros estándares. Comparativa

UNIDAD DIDÁCTICA 4. DESARROLLO DEL SOFTWARE DE GESTIÓN DE SISTEMAS

1. Análisis de especificaciones para el desarrollo de software de gestión de sistemas:
2. Identificación de los componentes necesarios según las especificaciones
3. Análisis de los componentes reutilizables
4. Análisis de la integración de los componentes en la arquitectura del sistema
5. Identificación de los modelos funcionales y de datos de los componentes

6. Técnicas de programación presentes en lenguajes de uso común aplicables al desarrollo de software de gestión de sistemas:
7. Programación estructurada:
8. Tipos primitivos y estructurados
9. Variables. Ámbito de utilización
10. Operadores aritméticos y lógicos
11. Estructuras de control. Bucles, condicionales y selectores
12. Funciones y procedimientos. Parámetros por valor y referencia.
13. Recursividad
14. Programación de elementos básicos: cadenas, fechas y ficheros.
15. Conversiones de tipos
16. Manejo de errores (excepciones)
17. Lenguajes estructurados de uso común
18. Programación orientada a objetos:
19. Clases y objetos
20. Herencia, polimorfismo y sobrecarga dinámica de métodos
21. Propiedades: selectores (get), modificadores (set) y referencias (let)
22. Lenguajes orientados a objetos de uso común
23. Técnicas de programación de software de gestión de sistemas:
24. Reutilización de código.
25. Uso de librerías del sistema
26. Llamadas a utilidades y aplicaciones del sistema
27. Técnicas específicas aplicables a los servicios básicos del sistema:
28. Programación de la gestión de los procesos: multitarea, control de bloqueos(deadlock) y comunicación entre procesos
29. Programación de la gestión de memoria: jerarquías de memoria, paginación de memoria, segmentación de memoria, intercambio(swapping), compartición de memoria, seguridad y memoria virtual
30. Programación de los sistemas de archivos: acceso a archivos y directorios, atributos y mecanismos de protección
31. Programación de los sistemas de entrada y salida: gestión de interrupciones, acceso directo a memoria (DMA) , puertos de entrada/salida y asignación de memoria
32. Programación de la seguridad: control de variables, control de desbordamiento de búferes , aserciones, precondiciones y post-condiciones.
33. Técnicas de optimización

34. Control de calidad del desarrollo del software de gestión de sistemas:
35. Métricas aplicables
36. Verificación de requisitos
37. Proceso de mejora continua
38. Herramientas de uso común para el desarrollo de software de sistemas:
39. Editores orientados a lenguajes de programación
40. Compiladores y enlazadores
41. Generadores de programas
42. Depuradores
43. De prueba y validación de software
44. Optimizadores de código
45. Empaquetadores
46. Generadores de documentación de software
47. Despliegue de software:
48. Gestores y repositorios de paquetes. Versionado y control de dependencias
49. Distribución de software
50. Gestores de actualización de software
51. De control de versiones
52. Entornos integrados de desarrollo (IDE) de uso común:
53. Específicos de sistemas Windows
54. Específicos de sistemas Unix
55. Multiplataforma

UNIDAD FORMATIVA 2. DESARROLLO DE COMPONENTES SOFTWARE PARA EL MANEJO DE DISPOSITIVOS (DRIVES)

UNIDAD DIDÁCTICA 1. EL NÚCLEO DEL SISTEMA OPERATIVO

1. Arquitectura general del núcleo
2. Subsistemas del núcleo:
3. Gestión de procesos
4. Gestión de memoria
5. Sistemas de ficheros
6. Control de dispositivos
7. Comunicaciones

8. Aspectos de seguridad sobre el desarrollo de elementos del núcleo
9. Consideraciones sobre compatibilidad de versiones del núcleo

UNIDAD DIDÁCTICA 2. PROGRAMACIÓN DE CONTROLADORES DE DISPOSITIVO

1. Funcionamiento general de un controlador de dispositivo
2. Principales tipos de controladores de dispositivo:
3. Carácter
4. Bloque
5. Paquete
6. Técnicas básicas de programación de controladores de dispositivos
7. Estructuras básicas de datos de dispositivos
8. Gestión de errores de dispositivos
9. Gestión de memoria de dispositivos
10. Control de interrupciones
11. Gestión de puertos de entrada y salida
12. Uso de Acceso directo a memoria (DMA) y buses
13. Técnicas de depuración y prueba:
14. Impresión de trazas
15. Monitorización de errores
16. Técnicas específicas de depuración de controladores en sistemas operativos de uso común:
17. Windows
18. Unix
19. Aplicación de estándares de calidad del software al desarrollo de controladores de dispositivos
20. Compilación y carga de controladores de dispositivos
21. Distribución de controladores de dispositivo
22. Particularidades en el desarrollo de dispositivos en sistemas operativos de uso común:
23. Sistemas Windows
24. Sistemas Unix
25. Modos de instalación de controladores de dispositivo en sistemas operativos de uso común. Dispositivos Plug Play:
26. Instalación de dispositivos en Windows
27. Instalación de dispositivos en Sistemas Unix

28. Herramientas:
29. Entornos de desarrollo de controladores de dispositivo en sistemas operativos de uso común
30. Herramientas de depuración y verificación de controladores de dispositivos
31. Documentación de manejadores de dispositivo:
32. Elaboración de especificaciones técnicas siguiendo directrices específicas de sistemas operativos de uso común
33. Elaboración de manual de instalación
34. Elaboración de manual de uso