Paraninfo

UF1844 - Desarrollo de aplicaciones web en el entorno servidor

Editorial: Paraninfo
Autor: JOSE LUIS BERENGUEL GÓMEZ
Clasificación: Certificados Profesionales > Informática y Comunicaciones
Tamaño: 17 x 24 cm.
Páginas: 268
ISBN 13: 9788428363877
ISBN 10: 8428363870
Precio sin IVA: $ 100947.00 COP
Precio con IVA: $ 100947.00 COP
Fecha publicacion: 28/08/2024

Sinopsis

El proceso de desarrollo del software es una parte de la Ingeniería Informática que incluye la planificación, desarrollo y seguimiento del proyecto para el aseguramiento de la calidad empleando estándares de la industria y las pruebas de software, así como el uso de herramientas informáticas que automaticen y apoyen estas y otras tareas presentes en el desarrollo de software.
A través de este libro aprenderemos a crear componentes software siguiendo el paradigma de programación orientada a objetos empleando para ello el lenguaje Java. Además, también se introduce en la creación de aplicaciones web utilizando la tecnología Jakarta Enterprise Edition (Jakarta EE). Cada capítulo se complementa con actividades prácticas cuyas soluciones están disponibles en www.paraninfo.es.
Los contenidos del libro se corresponden con los de la UF1844 *Desarrollo de aplicaciones web en el entorno servidor,* incardinada en el MF0492_3 *Programación web en el entorno servidor,* del certificado IFCD0210 *Desarrollo de aplicaciones con tecnologías web* regulado por el RD 1531/2011, de 31 de octubre, y modificado por el RD 628/2013, de 2 de agosto.

José Luis Berenguel Gómez es Doctor cum laude por la Universidad de Almería y profesor de ciclos formativos con 20 años de experiencia. Además de la Informática, sus aficiones son el deporte y la montaña.

Indice

1. El proceso de desarrollo de software

1.1. Modelos del ciclo de vida del software

1.1.1. En cascada (waterfall)

1.1.2. Iterativo

1.1.3. Incremental

1.1.4. En V

1.1.5. Basado en componentes (CBSE)

1.1.6. Desarrollo rápido (RAD)

1.1.7. Ventajas e inconvenientes. Pautas para la selección de la metodología más adecuada

1.2. Análisis y especificación de requisitos

1.2.1. Tipos de requisitos

1.2.2. Modelos para el análisis de requisitos

1.2.3. Documentación de requisitos

1.2.4. Validación de requisitos

1.2.5. Gestión de requisitos

1.3. Diseño

1.3.1. Modelos para el diseño de sistemas

1.3.2. Diagramas de diseño. El estándar UML

1.3.3. Documentación

1.4. Implementación. Conceptos generales de desarrollo de software

1.4.1. Principios básicos del desarrollo de software

1.4.2. Técnicas de desarrollo de software

1.5. Validación y verificación de sistemas

1.5.1. Planificación

1.5.2. Métodos formales de verificación

1.5.3. Métodos automatizados de análisis

1.6. Pruebas de software

1.6.1. Tipos

1.6.2. Pruebas funcionales (BBT)

1.6.3. Pruebas estructurales (WBT)

1.6.4. Comparativa. Pautas de utilización

1.6.5. Diseño de pruebas

1.6.6. Ámbitos de aplicación

1.6.7. Pruebas de componentes

1.6.8. Pruebas de sistemas

1.6.9. Automatización de pruebas. Herramientas

1.6.10. Estándares sobre pruebas de software

1.7. Calidad del software

1.7.1. Principios de calidad del software

1.7.2. Concepto de métrica y su importancia en la medición de la calidad

1.7.3. Métricas y calidad del software

1.7.4. Principales métricas en las fases del ciclo de vida software

1.7.5. Estándares para la descripción de los factores de calidad

1.7.6. ISO 9126

1.7.7. Otros estándares. Comparativa

1.8. Herramientas de uso común para el desarrollo de software

1.8.1. Editores orientados a lenguajes de programación

1.8.2. Compiladores y enlazadores

1.8.3. Generadores de programas

1.8.4. Depuradores

1.8.5. De prueba y validación de software

1.8.6. Optimizadores de código

1.8.7. Empaquetadores

1.8.8. Generadores de documentación de software

1.8.9. Gestores y repositorios de paquetes. Versionado y control de dependencias

1.8.10. De distribución de software

1.8.11. Gestores de actualización de software

1.8.12. De control de versiones

1.8.13. Entornos integrados de desarrollo (IDE) de uso común

1.9. Gestión de proyectos de desarrollo de software

1.9.1. Planificación de proyectos

1.9.2. Control de proyectos

1.9.3. Ejecución de proyectos .

1.9.4. Herramientas de uso común para la gestión de proyectos

Ejercicios

Ejercicios de comprobación

Ejercicios de aplicación

2. La orientación a objetos

2.1. Principios de la orientación a objetos. Comparación con la programación estructurada

2.1.1. Ocultación de información (information hiding)

2.1.2. El tipo abstracto de datos (ADT). Encapsulado de datos

2.1.3. Paso de mensajes

2.2. Clases de objetos

2.2.1. Atributos, variables de estado y variables de clase

2.2.2. Métodos. Requisitos e invariantes

2.2.3. Gestión de excepciones

2.2.4. Agregación de clases

2.3. Objetos

2.3.1. Creación y destrucción de objetos

2.3.2. Llamada a métodos de un objeto

2.3.3. Visibilidad y uso de las variables de estado

2.3.4. Referencias a objetos

2.3.5. Persistencia de objetos

2.3.6. Optimización de memoria y recolección de basura (garbage collection)

2.4. Herencia

2.4.1. Concepto de herencia. Superclases y subclases

2.4.2. Herencia múltiple

2.4.3. Clases abstractas

2.4.4. Tipos de herencia

2.4.5. Polimorfismo y enlace dinámico (dynamic binding)

2.4.6. Directrices para el uso correcto de la herencia

2.5. Modularidad

2.5.1. Librería de clases. Ámbito de utilización de nombres

2.5.2. Ventajas de la utilización de módulos o paquetes

2.6. Genericidad y sobrecarga

2.6.1. Concepto de genericidad

2.6.2. Concepto de sobrecarga. Tipos de sobrecarga

2.6.3. Comparación entre genericidad y sobrecarga

2.7. Desarrollo orientado a objetos

2.7.1. Lenguajes de desarrollo orientado a objetos de uso común

2.7.2. Herramientas de desarrollo

2.8. Lenguajes de modelización en el desarrollo orientado a objetos

2.8.1. Uso del lenguaje unificado de modelado (UML) en el desarrollo orientado a objetos

2.8.2. Diagramas para la modelización de sistemas orientados a objetos

Ejercicios

Ejercicios de comprobación

Ejercicios de aplicación

3. Arquitecturas web

3.1. Concepto de arquitectura web

3.1.1. Evolución de la web

3.1.2. Seguridad web

3.2. El modelo de capas

3.3. Plataformas para el desarrollo en las capas servidor

3.3.1. Plataformas en la capa de negocio

3.3.2. Plataformas en la capa de datos

3.3.3. Plataformas en la nube

3.4. Herramientas de desarrollo orientadas a servidor de aplicaciones web

3.4.1. Tipos de herramientas

3.4.2. Extensibilidad. Instalación de módulos

3.4.3. Técnicas de configuración de los entornos de desarrollo, preproducción y producción

3.4.4. Funcionalidades de depuración

Ejercicios

Ejercicios de comprobación

Ejercicios de aplicación

4. Lenguajes de programación de aplicaciones web en el lado servidor

4.1. Características de los lenguajes de programación web en servidor

4.2. Tipos y características de los lenguajes de uso común

4.2.1. Interpretados orientados a servidor

4.2.2. Lenguajes de cliente interpretados en servidor

4.2.3. Lenguajes compilados

4.3. Criterios en la elección de un lenguaje de programación web en servidor. Ventajas e inconvenientes
4.4. Características generales
4.4.1. Tipos de datos
4.4.2. Clases
4.4.3. Operadores básicos. Manipulación de cadenas de caracteres
4.4.4. Estructuras de control. Bucles y condicionales
4.4.5. Módulos o paquetes
4.4.6. Herencia
4.4.7. Gestión de bibliotecas (libraries)
4.5. Gestión de la configuración
4.5.1. Configuración de descriptores
4.5.2. Configuración de ficheros
4.6. Gestión de la seguridad
4.6.1. Conceptos de identificación, autenticación y autorización
4.6.2. Técnicas para la gestión de sesiones
4.7. Gestión de errores
4.7.1. Técnicas de recuperación de errores
4.7.2. Programación de excepciones
4.8. Transacciones y persistencia
4.8.1. Acceso a bases de datos. Conectores
4.8.2. Estándares para el acceso a bases de datos
4.8.3. Gestión de la configuración de acceso a bases de datos .
4.8.4. Programación de transacciones
4.8.5. Acceso a directorios y otras fuentes de datos
4.9. Componentes en servidor. Ventajas e inconvenientes en el uso de contenedores de componentes
4.10. Modelos de desarrollo. El modelo vista controlador
4.10.1. Modelo: programación de acceso a datos
4.10.2. Vista: desarrollo de aplicaciones en cliente. Eventos e interfaz de usuario
4.10.3. Programación del controlador
4.10.4. Documentación del software. Inclusión en código fuente. Generadores de documentación
Ejercicios
Ejercicios de comprobación
Ejercicios de aplicación
Bibliografía