

# PROGRAMACIÓN DE LOS CICLOS FORMATIVOS DE FORMACIÓN PROFESIONAL

## INFORMÁTICA Y COMUNICACIONES

### **TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB**

#### **Módulo 5: Entorno de desarrollo**

# ENTORNOS DE DESARROLLO

## OBJETIVOS

- 1.- Reconocer los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.
2. Evaluar entornos integrados de desarrollo, analizar sus características para editar código fuente y generar ejecutables.
3. Verificar el funcionamiento de programas diseñando y realizando pruebas.
4. Optimizar código empleando las herramientas disponibles en el entorno de desarrollo.
5. Generar diagramas de clases, valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.
6. Generar diagramas de comportamiento, valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.

# **ENTORNOS DE DESARROLLO**

## **CONTENIDOS**

### **I - DESARROLLO DEL SOFTWARE**

- 1.- PROGRAMA INFORMÁTICO
- 2.- OBTENCIÓN DE CÓDIGO EJECUTABLE
- 3.- INGENIERÍA DEL SOFTWARE
- 4.- FASES DEL PROCESO DE DESARROLLO DEL SOFTWARE
- 5.- METODOLOGÍA DEL DESARROLLO DE SOFTWARE
- 6.- RELACIÓN DEL SOFTWARE CON LOS COMPONENTES DEL SISTEMA
- 7.- ROLES QUE INTERACTUAN EN EL PROCESO DE DESARROLLO DE SOFTWARE
- 8.- ARQUITECTURAS DE SOFTWARE
- 8.1- PATRONES DE DISEÑO O DE DESARROLLO
- 8.2- DESARROLLO EN TRES CAPAS
- 9.- ENTORNOS DE DESARROLLO Y ENTORNOS DE PRODUCCIÓN

### **II – INSTALACIÓN Y USO DE ENTORNOS DE DESARROLLO**

- 1.- FUNCIONES DE UN ENTORNO DE DESARROLLO
- 2.- COMPONENTES DE UN ENTORNO DE DESARROLLO
- 3.- ENTORNOS DE DESARROLLO LIBRES Y COMERCIALES
- 4.- INSTALACION DE UN ENTORNO DE DESARROLLO
- 5.- MECANISMO DE ACTUACION DE UN ENTORNO DESARROLLO
- 6.- USO BÁSICO DE UN ENTORNO DE DESARROLLO: CARACTERÍSTICAS
- 6.1- EDICCIÓN DE PROGRAMAS
- 6.2- GENERACIÓN DE EJECUTABLES
- 7.- USO DE HERRAMIENTAS CASE EN EL DESARROLLO DEL SOFTWARE

### **III – DISEÑO Y REALIZACIÓN DE PRUEBAS**

- 1.- DEFINICIONES
- 2.-CICLO COMPLETO DE LAS PRUEBAS
- 3.- TÉCNICAS DE DISEÑO DE CASOS DE PRUEBA
- 3.1- PRUEBAS ESTRUCTURALES O DE CAJA BLANCA
- 3.2- PRUEBAS FUNCIONALES O PRUBAS DE CAJA NEGRA
- 3.2.1- PARTICIONES O CLASES DE EQUIVALENCIA

- 3.2.2- ANÁLISIS DE VALORES LÍMITE (AVL)
- 3.2.3- CONJETURA DE ERRORES
- 3.3- PRUEBAS ALEATORIAS
- 4.- ESTRATEGIAS DE APLICACIÓN DE LAS PRUEBAS
- 4.1- PRUEBAS UNITARIAS
- 4.2- PRUEBAS DE INTEGRACIÓN
- 4.3- PRUEBAS DEL SISTEMA
- 4.4- PRUEBAS DE ACEPTACIÓN

## **IV – OPTIMIZACIÓN Y DOCUMENTACIÓN**

- 1.- REFACTORIZACIÓN
- 1.1- TABULACIÓN
- 1.2- PATRONES DE REFACTORIZACIÓN MÁS USUALES
- 1.3- MALOS OLORES
- 1.4- REFACTORIZACIÓN Y PRUEBAS
- 1.5- HERRAMIENTAS
- 2.- CONTROL DE VERSIONES
- 2.1- REPOSITORIOS
- 2.2- HERRAMIENTAS
- 3.- DOCUMENTACIÓN
- 3.1- USO DE COMENTARIOS
- 3.2- HERRAMIENTAS

## **V – ELABORACIÓN DE DIAGRAMAS DE CLASE**

- 1.- INTRODUCCIÓN A LA ORIENTACIÓN A OBJETOS
- 2.- INTRODUCCIÓN A UML
- 2.1.- TIPOS DE DIAGRAMAS UML
- 2.2.- DIAGRAMAS DE CLASE
- 3.- HERRAMIENTAS PARA EL DISEÑO DE DIAGRAMAS
- 3.1- GENERACIÓN DE CÓDIGO A PARTIR DE DIAGRAMAS DE CLASE
- 3.2- INGENIERÍA INVERSA

## **VI – ELABORACIÓN DE DIAGRAMAS DE COMPORTAMIENTO**

- 1.- TIPOS Y CAMPO DE APLICACIÓN
- 2.- DIAGRAMAS DE CASO DE USO
- 3.- HERRAMIENTAS PARA LA ELABORACIÓN DE DIAGRAMAS
- 4.- DIAGRAMAS DE SECUENCIA
- 5.- INGENIERÍA INVERSA

# **CAPACIDADES Y APRENDIZAJES MINIMOS PARA APROBAR**

- 1.- Reconocer los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.
2. Saber utilizar entornos integrados de desarrollo, analizar sus características para editar código fuente y generar ejecutables.
3. Saber verificar el funcionamiento de programas diseñando y realizando pruebas.
4. Saber optimizar código empleando las herramientas disponibles en el entorno de desarrollo.
5. Conocer cómo generar diagramas de clases, valorar su importancia en el desarrollo de aplicaciones y emplear las herramientas disponibles en el entorno.
6. Saber generar diagramas de comportamiento, valorar su importancia en el desarrollo de aplicaciones y emplear las herramientas disponibles en el entorno.

## **CRITERIOS Y MOMENTOS DE EVALUACIÓN**

- 1.- Se valorará diariamente el trabajo aportado por el alumno al realizar ejercicios prácticos/teóricos que correspondan.
- 2.- Periódicamente se realizarán pruebas especiales escritas o en el ordenador coincidiendo generalmente con la finalización del estudio de cada unidad didáctica.
- 3.- Dado que un porcentaje muy elevado del módulo es procedimental se valorará continuamente comportamiento, postura activa para el trabajo, atención y asistencia diaria a clase.
- 4.- La asistencia a clase es obligatoria y la ausencia de más de un 20% supondrá la pérdida del derecho a la evaluación continua. En dicho caso el alumno sólo tendrá derecho a la realización de un examen.
- 5.- Se sugiere en cada unidad unos criterios de evaluación encaminados a evaluar con un 80% las pruebas escritas, con un 10% los trabajos individuales o en grupo y un 10% las pruebas orales y de participación del alumno en clase.  
Las notas obtenidas por cada alumno deben unificarse en una calificación final, mediante una media ponderada de las mismas, en la que se pondere con un coeficiente adecuado para valorar la importancia de las diferentes unidades, en la calificación final.
- 6.- Las recuperaciones serán en las convocatorias extraordinarias finales.