

CERTIFICADO

DESARROLLO DE APLICACIONES CON TECNOLOGÍA WEB

MODULO PROGRAMACION WEB EN ENTORNO SERVIDOR

- **Desarrollo de aplicaciones web en entorno servidor**
- **Acceso a datos en aplicaciones web del entorno servidor**
- **Desarrollo de aplicaciones web distribuidas**



TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

Partimos que en informática, debemos distinguir entre **hardware y software**. Que el Hardware es el elemento físico de la informática y el software el elemento lógico.

Aunque ambas están unidas , son muy diferentes.

Y a su vez cada software es muy diferente dependiendo de la función que tenga.

Construir un producto Software implica:

1. Entender para qué es necesario
2. Diseñar el producto para que cumpla los requisitos
3. Requiere que el equipo de desarrollo esté coordinado

Por todo esto **necesitamos procesos bien definidos a la hora de desarrollar un software. Vamos a ver los modelos más utilizados y la actividades que se deben realizar en cada uno de ellos.**

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

Partimos que en informática, debemos distinguir entre **hardware y software**. Que el Hardware es el elemento físico de la informática y el software el elemento lógico.

Aunque ambas están unidas , son muy diferentes.

Y a su vez cada software es muy diferente dependiendo de la función que tenga.

Construir un producto Software implica:

1. Entender para qué es necesario
2. Diseñar el producto para que cumpla los requisitos
3. Requiere que el equipo de desarrollo esté coordinado

Por todo esto **necesitamos procesos bien definidos a la hora de desarrollar un software. Vamos a ver los modelos más utilizados y la actividades que se deben realizar en cada uno de ellos.**



TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

1 MODELOS DEL CICLO DE VIDA DEL SOFTWARE

Se trata de estrategias de desarrollo que ayudan a organizar las diferentes etapas y actividades del ciclo de vida del software.

ayudan al control y a la coordinación del proyecto

1. En cascada

el más básico de todos los modelos
enfoca el desarrollo de software, así:

1. a través de una secuencia simple de fases con un proceso secuencial . Fluye como en cascada
2. Ordena rigurosamente las etapas del ciclo de vida del software : el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior
3. Cada fase tiene una meta bien definida

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

Fases de modelo en cascada

◆ Análisis de requerimientos

- ◆ El **qué**.
- ◆ Se especifica la información sobre la cual el software se a a desarrollar.

◆ Diseño

- ◆ comienza una vez finalizada la etapa anterior.
- ◆ Es el **cómo** el software va a satisfacer dichos requisitos

◆ Implementación

- ◆ comienza una vez finalizada la etapa anterior.
- ◆ Fase en la que se codifica el software

◆ Pruebas

- ◆ Fase de validación y verificación

◆ Mantenimiento

- ◆ comienza una vez finalizada que el software se ha implantado en el cliente y lo ha comenzado a utilizar.
- ◆ En esta fase se realizan modificaciones por errores, adecuaciones, nuevos requerimientos

:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

1 MODELOS DEL CICLO DE VIDA DEL SOFTWARE

2. Iterativo

- ◆ Nace para reducir el riesgo que surge entre las necesidades del usuario y el producto final
- ◆ Consiste en la iteración de varios ciclos de vida en cascada
- ◆ Cada iteración sirve para hacer una mejora. Momento en que el cliente evalúa el producto, y lo corrige o propone mejoras. Así hasta que satisfaga totalmente las necesidades del cliente

3. Incremental

- ◆ Combina los dos anteriores
- ◆ Incrementa a cada paso las funcionalidades del programa
- ◆ Aplica secuencias lineales de forma escalonada
- ◆ Cada secuencia lineal produce un incremento del software
- ◆ Muy útil por que reduce los riesgos en el desarrollo de sistemas largos y complejos, ya que el software se va construyendo por partes
- ◆ Principal ventaja: permite crear cada vez versiones más complejas de software, construyendo versiones sucesivas de un producto: Se crea una primera versión que es utilizada por el usuario, y según los requerimientos especificados de este usuario se crea una segunda versión

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

4. En V

- ◆ Nace para terminar con algunos problemas del método de cascada tradicional: es decir cuando los errores se encuentran hacia el final cuando comienzan las pruebas.
- ◆ Así propone que la pruebas comiencen lo antes posible . Integrando la verificación y validación en cada ciclo de vida
 - ◆ La parte izquierda de la V representa la descomposición de los requisitos y la creación de las especificaciones del sistema.
 - ◆ El lado derecho de la V representa la integración de partes y su verificación

5. Basado en componentes (CBSE)

Modelo que construye el software con la ayuda de productos prefabricados de software , que reciben el nombre de componentes y generalmente son comerciales

Cada componente tiene una funcionalidad

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

5. Desarrollo rápido(RAD)

Para responder a las necesidades de entregar sistemas muy rápidamente.(No es apropiado para todos los proyectos)

- ◆ el éxito lo determina: el alcance el tamaño y las circunstancias
- ◆ Es una fusión de varias técnicas estructuradas
 - ◆ Requiere el uso interactivo de técnicas estructuradas y prototipos para definir los requisitos de usuario
 - ◆ El desarrollador primer construye modelos de datos y modelos de procesos de negocio preliminares de los requisitos.
 - ◆ Los prototipos ayudan al analista y a los usuarios a verificar tales requisitos y a refinar los modelos de datos y procesos

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

5. Ventajas e inconvenientes. Cual es la metodologia más adecuada?

modelo	ventaja	inconvenientes	adecuación
cascada	Simple y fácil de usar puesto que cada fase tiene plazos de entrega y revisiones definidas	Conseguir que el cliente establezca al principio todos los requisitos necesarios. Los resultados o mejoras no son visibles inmediatamente	para proyectos no cambiantes, estables. Proyectos pequeños donde los requisitos están bien entendidos

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

5. Ventajas e inconvenientes. Cual es la metodologia más adecuada?

modelo	ventaja	inconvenientes	adecuación
iterativo	No hace falta que los requisitos están bien definidos al principio. Permite gestionar mejor los riesgos y entregas	A veces el no tener los requisitos bien definidos desde el principio puede dar problemas relacionados con la arquitectura	Proyectos muy cambiantes
incremental	genera software operativo de forma rápida. Mas flexible lo cual reduce el coste Es más fácil probar y depurar Es más fácil gestionar riesgos	Requiere gran experiencia para definir los incrementos. Puede dar problemas con arquitectura por no	Proyectos muy cambiantes y con requisitos no definidos

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

5. Ventajas e inconvenientes. Cual es la metodologia más adecuada?

modelo	ventaja	inconvenientes	adecuación
En V	Simple y fácil de utilizar Cada fase tiene plazos de entrega específicos Mayor probabilidad de éxito que el modelo en cascada puesto que se desarrolla planes de prueba en etapas tempranas	Modelo rígido igual que el modelo en cascada. Poca flexibilidad NO proporciona caminos claros para problemas encontrados durante las fases de pruebas	Proyectos pequeños donde los requisitos son entendidos fácilmente
CBSE	Permite reutilización de software	Dependen de las limitaciones de los componentes. Estos pueden ser caros , obsoletos	Equipos de desarrollo con una amplia biblioteca de componentes

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Modelos de ciclo de vida del Software

5. Ventajas e inconvenientes. Cual es la metodologia más adecuada?

modelo	ventaja	inconvenientes	adecuación
RAD	Velocidad de desarrollo Visibilidad temprana gracias a técnicas de prototipado Mayor flexibilidad que otros modelos Ciclos de desarrollo mas cortos	Escalabilidad reducida Más difícil de evaluar el proceso porque no hay hitos clásicos	Proyectos con poco tiempo de desarrollo

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Análisis y especificación de requisitos

¿Qué es un requisito?

- Aquello que necesita un usuario para conseguir un objetivo determinado.
- Tener claros los requisitos es básico para crear un software, de ahí que se deban plasmar en un documento llamado Especificación de Requisitos
- Normalizados por la IEEE Std 830-1998

Tipos de requisitos

- **Requisitos de usuario** **Expresados** verbalmente por los usuarios
- **Requisitos del sistema** Componentes que el sistema debe tener para realizar determinadas tareas.
- **Requisitos funcionales** Servicios o funciones que el programa debe proporcionar
- **Requisitos no funcionales** Propiedades o cualidades que el producto debe tener. Pueden clasificarse en :
 - **Requisitos de rendimiento**
 - **Requisitos de seguridad**
 - **Requisitos de frecuencia de tratamiento**

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Análisis y especificación de requisitos

Modelos para el análisis de requisitos

Cualquier modelo de análisis debe cumplir tres objetivos primarios

- Describir lo que requiere el cliente
- Establecer una base para la creación de un diseño de software
- Definir un conjunto de requisitos que pueda validarse una vez construido el software.

Técnicas para realizar la recopilación de o requerimientos :

- **Entrevistas** Utilizadas para recopilar información de los interesados.
Importante utilizar preguntas abiertas, porque la predisposición y experiencia s de la persona entrevistada influirá en la obtención de resultados
- **Análisis de documentos** definen la razón de ser del proyecto
- **Tormenta de ideas** técnica eficaz porque las ideas más creativas y efectivas son , a menudo , el resultado de la combinación de ideas aparentemente inconexas.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Análisis y especificación de requisitos

Documentación de requisitos

Se debe hablar de :

- especificación de requerimientos
- Definición de requerimientos

Ambos cubren exactamente lo mismo , la diferencia la marca el lector.

Normalmente la definición de los requerimientos está redactada en lenguaje natural, mientras que la especificación de los requerimientos se recata de una forma más técnica.

Validación de requisitos

Es importante asegurarse de que los requisitos están completos, son exactos y consistentes.

Verificaciones que deben llevarse a cabo durante el proceso de validación:

- Verificación de validez. Que sea lo que realmente pedía el usuario
- Verificación de consistencia. No debe haber restricciones o descripciones contradictorias
- Verificación de completitud. Cumplan con todas las funciones propuestas
- Verificación de realismo. Asegurar que los requerimientos pueden cumplirse teniendo en cuenta la tecnología existente, presupuesto y tiempo disponible
- Verificabilidad deben redactarse de tal forma que sean verificables

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Análisis y especificación de requisitos

Técnicas de validación de requerimientos:

- Revisiones de requerimientos y construcción de prototipos
 - proceso manual en el que interviene tanto el cliente como el personal involucrado en el desarrollo del sistema.
 - Puede ser formal o informal
 - Fin de verificar que el documento no presente anomalías ni omisiones
- Generar casos de prueba

Gestión de requisitos

El proceso de organizar y llevar a cabo los cambios en los requerimientos se llama gestión de requisitos o requerimientos

Objetivo del análisis : reconocer los elementos básico del sistema, tal y como lo percibe el usuario/cliente

El analista debe contactar al usuario con la empresa que vaya a desarrolla.

Cuando los usuario finales experimentan un sistema, descubren nuevas necesidades y prioridades

En definitiva la gestión de requisitos es el proceso de comprender y controlas los cambios en los requerimientos del sistema

ventaja: los cambios propuestos son tratados de forma consistente y los cambios en el documento de requerimientos se hace de forma controlada

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Diseño

3 . DISEÑO

Se trata de desarrollar las directrices propuestas, que tenga más posibilidades de satisfacer los objetivos planteados.

Proceso de diseño:

- ◆ Se suele hacer de forma descendente
 - ◆ Diseño de alto nivel
 - ◆ Diseño e implementación de cada uno de los subsistemas:
 - ◆ Especificación consistente y completa del subsistema de acuerdo con los objetivos establecidos en el análisis
 - ◆ Desarrollo según la especificación
 - ◆ Prueba
 - ◆ Integración de todos los subsistemas
 - ◆ Validación de diseño

Conclusión: el diseño de sistemas es el arte de definir la arquitectura de hardware y software , componentes, módulos y datos de un sistema de computo para satisfacer ciertos requerimientos. Es la etapa posterior al análisis de sistemas.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Diseño

- ◆ Sus objetivos comunes son:
 - ◆ Acercar el modelo de análisis al modelo de implementación
 - ◆ Identificar requisitos no funcionales y restricciones en relación a lenguajes de programación , reutilización de componentes, sistema operativos, tecnologías de distribución, concurrencia, bases de datos, interfaces de usuario, gestión de transacciones...
 - ◆ Descomponer el modelo de análisis en subsistemas que puedan desarrollarse en paralelo
 - ◆ Definir la interfaz de cada subsistema
 - ◆ Derivar una representación arquitectónica del sistema

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Diseño

- ◆ **Diagramas de diseño. El estándar UML (Unifed Modeled Lenguaje)**
EL UML es un lenguaje unificado de modelado , que permite capturar información acerca de la estructura estática y el comportamiento dinámico de un sistema, e incluye conceptos, semánticos, , notación y reglas de creación de diferentes tipos de diagramas
- ◆ Se propone el uso de dos tipos de programas
 - ◆ EL DIAGRAMA DE CASOS DE USO, que ayuda a determinar la funcionalidad del sistema desde la perspectiva del usuario
 - ◆ EL DIAGRAMA DE INTERACCION O DIAGRAMA DE SECUENCIA, donde se muestran las interacciones entre objetos mediante transferencias de mensajes entre objetos o subsistemas

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Implementación.

Conceptos generales de desarrollo de software

- ◆ **Implementación**
Etapa en la que se comienza a codificar el diseño realizado en la etapa anterior.

Principios básicos del desarrollo del software:

- ◆ **Rigor y formalidad**
una aproximación rigurosa produce productos más fiables
La formalidad puede ser la base para la mecanización del proceso
- ◆ **Separación de intereses**
se debe intentar separar los temas que no están íntimamente relacionados entre si
- ◆ **Modularidad**
Un sistema complejo debe ser dividido en partes
- ◆ **Abstracción**
los modelos que construimos para entender los fenómenos son abstracciones de la realidad.
El uso de modelos formales abstractos permite aproximarse mediante refinamientos a la solución final pudiendo demostrar que cada descripción verifica la anterior
- ◆ **Anticipación del cambio.** El software sufre cambios permanentemente
- ◆ **Generalidad** Siendo más general, la solución será mas reutilizable
- ◆ **Incrementabilidad** se identifica de forma temprana subconjuntos útiles de una aplicación para así obtener rápido feedback

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Implementación.

Conceptos generales de desarrollo de software

Técnicas de desarrollo de software

◆ **Técnicas para la recopilación de datos:**

Aquellas que se utilizan para determinar los requerimientos
Podemos destacar:

- ◆ La entrevista,
- ◆ Los cuestionarios
- ◆ La observación in situ y el análisis de documentos
- ◆ **Técnica de costo-beneficio:** es una técnica analítica que enumera y compara el costo neto de una intervención con los beneficios que surgen como consecuencia de aplicar dicha intervención.
- ◆ **Técnica de planificación y control de proyectos:** Es la técnica que permite realizar la planificación del proyecto definiendo los objetivos, el alcance el calendario del proyecto

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Validación y verificación de sistemas

- ◆ **Verificación:** conjunto de actividades que aseguran que el software implementa correctamente una función específica.
Técnica más utilizada : la revisión del SW
- ◆ **Validación:** se refiere a un conjunto diferente de actividades que aseguran que el software construido, se ajusta a los requisitos del cliente. La técnica más utilizada en la validación son las pruebas SW
- ◆ **Objetivos concretos**
 - ◆ Detectar y corregir los defectos tan pronto como sea posible
 - ◆ Disminuir os riesgos las desviaciones sobre los presupuestos y sobre el calendario o programa de tiempos del proyecto
 - ◆ Mejorar la calidad y fiabilidad del software
 - ◆ Mejorar la visibilidad de la gestión de proceso de desarrollo
 - ◆ Valorar rápidamente los cambios propuestos y sus consecuencias

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Validación y verificación de sistemas

◆ **Planificación** **definir claramente:**

- ◆ el objetivo,
- ◆ los tiempos,
- ◆ recursos,
- ◆ herramientas, y
- ◆ técnicas que se utilizarán
- ◆ Las etapas del ciclo de vida
- ◆ Que informas se emitirá
- ◆ Procedimientos de control que se llevarán a cabo para asegura que el proceso se realice correctamente

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Validación y verificación de sistemas

◆ Métodos formales de verificación

- ◆ Aserciones E/S. El programa se especifica mediante aserciones que relacionan las entras de salidas del programa
- ◆ Precondición más débil: Básicamente, consiste en dad una pos-condición POST, encontrar operando hacia atrás , un programa tal que se satisfaga en un amplio conjunto de situaciones
- ◆ Inducción estructural técnica de verificación formal que se basa en el principio de inducción matemática

◆ Métodos automatizados de análisis

- ◆ Herramientas de software que recorren el código fuente y detectan posibles anomalías y faltas. En general no requieren ejecución de código a analizar. Se trata de herramientas que complementan al compilador

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

- Las pruebas (testing) se realizan para evaluar la calidad de un producto y/o mejorarlo mediante la identificación de defectos y problemas.
Se trata de un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.
Al considerarse el software como un elemento del sistemas , y el coste que supone un fallo del propio sistemas, se esta incentivando las pruebas minuciosas y bien planificadas.

TIPOS

Cualquier producto de ingeniería puede probarse de una de estas dos formas, o de las dos a la vez:

- Conociendo la función específica para la que fue diseñado el producto, se van llevando a cabo pruebas que demuestren que cada función es completamente operativa al tiempo que se busca errores en cada función.
- Conociendo el funcionamiento del producto se pueden realizar pruebas que aseguren que todas las piezas encajan , o sea que la operación interna se ajusta a los requisitos

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

◆ PRUEBAS FUNCIONALES (BBT)

Las pruebas funcionales o pruebas de caja negra (*Black Box Testing*), son pruebas en las que el sistema a probar es visto como una caja negra, es decir, se desconoce su estructura interna y cualquier detalle de implementación. La prueba proporciona un conjunto de datos de entrada al sistema y comprueba que los datos proporcionados en la salida son los esperados.

◆ PRUEBAS ESTRUCTURALES (WBT)

Las pruebas estructurales o pruebas de caja blanca (*White Box Testing*) son una técnica de diseño de casos de prueba que se encarga de comprobar la estructura interna del software. Entre los aspectos que se tratan de comprobar están:

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

Aspectos que se tratan de comprobar:

- Si se han recorrido todas las rutas posibles dentro de un módulo.
- Si se han comprobado las decisiones lógicas para todas las combinaciones de valores verdadero y falso.
- Si se han ejecutado los bucles en sus valores frontera.
- Si se ha comprobado la validez de las estructuras de datos internas.

En la *Sección 1.6.5. Diseño de pruebas* se explican algunas técnicas de diseño de pruebas de caja blanca.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

💧 COMPARATIVA . PAUTAS DE UTILIZACION

Las pruebas de caja blanca están enfocadas a la estructura interna del programa, mientras que las pruebas de caja negra se centran en su comportamiento. Estas pruebas no son excluyentes y es probable que un error que no fue detectado por una de ellas, sea detectado por la otra.

Las pruebas de caja negra tienen como objetivo principal revisar que se cumplen los requisitos funcionales del programa, que no hay requisitos incompletos o que se han pasado por alto, que la interfaz no contiene errores, etc.

Las pruebas de caja blanca suelen realizarse antes en la fase de desarrollo que las pruebas de caja negra. Por ello, las pruebas que están más cercanas al código (pruebas de unidad y pruebas de integración) suelen ser de caja blanca, mientras que las pruebas que tratan de analizar el software en su conjunto (pruebas de validación y pruebas de sistemas) suelen ser de caja negra.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

Las pruebas no se aplican únicamente al código, sino que se debe elaborar un plan de pruebas que englobe el conjunto del proyecto (todas sus fases y documentación), lo que garantizará la calidad del producto final. El plan de pruebas debe tener en cuenta los siguientes aspectos:

- Qué se va a probar. Se deben identificar aquellos elementos del producto que se van a probar.
- Cómo se va a probar. Se debe escoger la estrategia o técnica más adecuada para probar los elementos. No tiene por qué ser única, es decir, diferentes elementos pueden requerir de diferentes técnicas de prueba.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

- Cuándo se va a probar. Se deben establecer hitos en el calendario del proyecto donde esté previsto realizar las pruebas.
- Análisis de las pruebas. Se deben tener claras cuáles serán las medidas de corrección a aplicar cuando una prueba falla. Dependerá del tipo de prueba y del momento en la fase de desarrollo en la que se realice.
- Finalización de las pruebas. Se deben tener claros los criterios para dar por finalizadas las pruebas. Este es un aspecto muy importante ya que puede determinar la calidad del producto final.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

💧 DISEÑO DE PRUEBAS

En esta sección se explicarán diferentes técnicas empleadas en el diseño de casos de prueba de caja negra y de caja blanca.

Pruebas de caja negra

Se explicarán las siguientes técnicas: partición de equivalencia, análisis de valor de frontera y prueba de array ortogonal.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

Partición de equivalencia

Este método de prueba de caja negra agrupa el dominio de datos de entrada en diferentes clases llamadas clases de equivalencia. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

Para identificar las clases de equivalencia se analiza la condición de entrada o salida y se divide en uno o más grupos. Hay dos tipos de clases de equivalencia: válidas e inválidas. Se pueden seguir las siguientes directrices para definir las clases de equivalencia:

- Si una condición de entrada especifica un rango, se define una clase de equivalencia válida y dos inválidas.
- Si una condición de entrada especifica un número de valores, se define una clase de equivalencia válida y dos inválidas.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

♦ **AMBITOS DE APLICACIÓN**

objetivo a probar durante la realización de las mismas.

En función del alcance de dichas pruebas se establecen 3 niveles o ámbitos de aplicación.

Partimos del nivel más específico al más general:

- ♦ Módulo único: pruebas unitarias
- ♦ Grupo de módulos : pruebas de integración o pruebas de componentes
- ♦ Sistemas completo: pruebas de sistemas

♦ **PRUEBAS DE SISTEMAS**

objetivo: ejercitar profundamente el sistema para verificar que se han integrado adecuadamente todos los elementos hardware y software del propio sistema entre si y con lo elementos s y h de otros sistemas con lo s que debería interactuar

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

Concretamente se debe comprobar:

- ◆ El cumplimiento de los requisitos funcionales establecidos. Pruebas de técnica caja negra
- ◆ El funcionamiento y rendimiento de las interfaces hardware, software y de usuario. Utiliza pruebas que verifiquen toda la información
- ◆ La adecuación de la documentación del usuario
- ◆ Rendimiento y respuesta en condiciones límite y de sobrecarga. Para lo que se diseñan pruebas que comprueben el rendimiento del sistema. Suelen llamarse pruebas de sobrecarga

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

Hay cuatro tipos de pruebas de sistemas:

- ◆ **Pruebas de recuperación.** Comprueban que el sistema es capaz de recuperarse de un fallo grave y seguir funcionando
- ◆ **Pruebas de seguridad.** Para protegerlo de accesos no deseados. El papel del diseñador del sistema es hacer que el coste de la entrada ilegal sea mayor que el valor de la información obtenida
- ◆ **Pruebas de resistencia** Se encargad de programas con situaciones anormales . Su objetivo es responder a la pregunta (a qué potencia puedo ponerlo a funcionar antes de que falle?
Para ello la prueba de resistencia ejecuta un sistema que produzca una demanda de recursos en cantidad, frecuencia o volúmenes anormales
- ◆ **Pruebas de rendimiento** abarca todos los pasos del proceso de prueba ya que deberíamos ser capaces de asegurar que se satisfacen los requisitos de rendimiento
Las pruebas de rendimiento suelen realizarse en paralelo con las de resistencia

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

PRUEBAS DE COMPONENTES:

Aunque los módulos o componentes de un programa funcionen bien ir separado es necesario probarlos.

Hay una tendencia a probar cada módulo por separado y luego se integran todos de una vez y se prueba el programa completo. Esta puede producir un gran número de errores y será prácticamente imposible identificar el módulo o módulos que los provocó

Por eso se debe aplicar la integración incremental, donde se combina el siguiente módulo que se debe probar con el conjunto de módulos que ya han sido probados. Así se va probando en pequeñas porciones y los errores son más fáciles de detectar.

Hay dos tipos ascendente y descendente

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

AUTOMATIZACIÓN DE PRUEBAS

Consiste en el uso de un software especial que ejecuta pruebas de manera controlada, presentando resultados y comparándolos con los resultados esperados.

Se recurre a frameworks tales como Junit (java) y Nunit (.Net)

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

Pruebas de software

ESTÁNDARES SOBRE PRUEBAS DE SOFTWARE

El actual ISO/IEC 29119, tiene como objetivo unificar estándares.
Consta de 6 partes:

1. Conceptos y definiciones donde se definen los conceptos y definiciones utilizada en la pruebas de software
2. Procesos que incluyen la evaluación de dichos procesos , donde se defienden cada uno para ser probados y las estrategia si aplicar en cada uno de ellos
3. Documentación que sirva para definir plantillas que pueden ser utilizadas para generar nueva documentación
4. Técnicas de pruebas que definen cada una de ellas, las medidas de cobertura, el alcance....
5. Pruebas basada en palabras claves que define el conjunto de palabras claves y las diferentes estrategias que se pueden aplicar
6. Técnicas de pruebas estáticas

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

CALIDAD DEL SOFTWARE

- ◆ **Calidad:** grado en que un conjunto de características inherentes cumple con los requisitos
- ◆ **PRINCIPIOS DE CALIDAD DEL SOFTWARE**
 - ◆ **Corrección** Capacidad de realizar las actividades requeridas
 - ◆ **Robustez** Capacidad de reacción ante imprevistos
 - ◆ **Eficiencia** capacidad de hacer uso adecuado de los recursos del sistema
 - ◆ **Portabilidad** facilidad de migración entre diferentes plataformas
 - ◆ **Integridad** capacidad del sistema para proteger sus componentes contra accesos no permitidos
 - ◆ **Facilidad de uso** facilidad para la realización de pruebas
 - ◆ **Extensibilidad** facilidad de adaptación ante nuevos requisitos
 - ◆ **Reutilización** Capacidad de que el software como bloque, actúe como componente dentro de un nuevo sistema

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

CALIDAD DEL SOFTWARE

♦ **METRICAS Y CALIDAD DEL SOFTWARE**

la métrica proporciona medidas que permiten evaluar esa calidad de manera objetiva.
Para controlar estas medidas se establecen modelos de calidad

♦ **CONCEPTO DE METRICA Y SU IMPORTANCIA EN LA MEDICION DE CALIDAD**

Métrica: medida de alguna propiedad del software

Una buena métrica debe ser simple, fáciles de calcular, de naturaleza empírica, objetivas, independientes del lenguaje de programación, fáciles de usar

♦ **PRINCIPALES MÉTRICAS EN LAS FASES DE CICLO DE VIDA DEL SOFTWARE**

- ♦ **Métricas en el modelo de análisis** intentan predecir el tamaño del sistema
- ♦ **Métricas de diseño** miden la arquitectura del sistema
- ♦ **Métricas de codificación** Ofrecen datos sobre la complejidad del código
- ♦ **Métricas de pruebas** evalúan el éxito de las pruebas, la cobertura de los requisitos y la tasa de eliminación de errores

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

CALIDAD DEL SOFTWARE

- ♦ **ESTÁNDARES PARA LA DESCRIPCION DE LOS FACTORES DE CALIDAD**
Recomendaciones a seguir para garantizar que el software , además de tener calidad, cumpla con los requisitos del cliente.
- ♦ Su uso significa que el proceso cumple con unos mínimos
 - ♦ Funcionalidad: adecuación, exactitud, la interoperabilidad, la seguridad y el cumplimiento funcional.
 - ♦ Fiabilidad: madurez, capacidad de recuperación, tolerancia a fallos.
 - ♦ Usabilidad: Los atributos que se evalúan con la comprensión, la operatividad y la capacidad de atracción
 - ♦ Eficiencia se evalúa el comportamiento en el tiempo y el comportamiento de los recursos
 - ♦ Mantenibilidad relacionado con la capacidad de modificar, corregir errores en un sistema software.
 - ♦ Portabilidad conjunto relacionados con la capacidad de un sistemas software para ser transferido desde una plataforma a otra
 - ♦ Calidad conjunto de atributos relacionados con la aceptación por parte del usuario final y del departamento de seguridad
- ♦ **OTROS ESTÁNDARES:** SPICE: PSP; CMME TSP; CMMI

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

HERRAMIENTAS DE USO COMÚN PARA EL DESARROLLO DEL SOFTWARE

♦ EDITORES ORIENTADOS A LENGUAJES DE PROGRAMACION

Herramienta principal de trabajo de un programador.
a veces están optimizados para diferentes tipos de proyectos o lenguajes
suelen incluir funciones espáticamente dedicadas a la programación: resultado de sintaxis, autoidentificación....

♦ COMPILADORES Y ENLAZADORES

programas que traducen un fichero de código fuente

Un compilador es un programa que permite traducir el código fuente de un programa en lenguaje de alto nivel otro lenguaje de nivel inferior (típicamente lenguaje de máquina).

Los enlazadores son programas que enlazan varios ficheros objeto en lenguaje binario para crear un único fichero , el ejecutable del programa

♦ GENERADORES DE PROGRAMAS

Herramientas que , a partir de ciertas entradas como datos y determinados elementos, generan el ejecutable del programa que queremos desarrollar . Uno de los más populares EMF(ECLIPSE Modelling Framework , el plug-in de Eclipse que permite a partir de una serie de modelos que el desarrollador va especificando, crear el código que ejecutará el programa

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

HERRAMIENTAS DE USO COMÚN PARA EL DESARROLLO DEL SOFTWARE

◆ **DEPURADORES**

Para depurar o corregir errores se encargan de ejecutar paso a paso el programa para detectar los errores que puedan existir .

Muy útiles para cuando el programa parece estar bien pero no obtiene el resultado esperado

el más importante: GDB, Actualmente todas las IDE incluyen uno o deberían incluirlo

◆ **DE PRUEBA Y VALIDACION DE SOFTWARE**

Herramientas que permiten automatizar la etapa de prueba y validación de software.

Ayudan al desarrollador a definir y ejecutar las diferentes pruebas a realizar

◆ las más utilizadas:

- ◆ Selenium compuesto por dos herramientas:

- ◆ Selenium IDE Permite crear casos e prueba para aplicaciones web

- ◆ Selenium WebDriver los ejecuta

- ◆ Jmeter permite realizar pruebas funcionales y de rendimiento para aplicaciones web.

- ◆ Testlink Permite gestionar y crear casos de prueba, organizarlos en planes de pruebas, realizar un seguimiento de los resultados, establecer trazabilidad con los requisitos, generar informes

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

HERRAMIENTAS DE USO COMÚN PARA EL DESARROLLO DEL SOFTWARE

OPTIMIZADORES DE CÓDIGO

Crean un nuevo código más compacto y eficiente, a partir del análisis del código generado por el programador: elimina sentencias que no se ejecutan nunca, simplifica expresiones aritméticas.

Los optimizadores de compilador se aplican generalmente mediante una secuencia de transformaciones de optimización , algoritmos que transforman un programa para producir otro con una salida semánticamente equivalente pero optimizada

EMPAQUETADORES

programa que se utiliza para reducir el tamaño de un archivo ejecutable generalmente, mediante la compresión del mismo.

Se utilizan empaquetadores para poder hacer la distribución del código ejecutable del programa

GENERADORES DE DOCUMENTACION DE SOFTWARE

herramientas que dan soporte a la etapa de análisis, diseño y codificación del ciclo de vida del software

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ Crear software es una actividad compleja, que involucra a mucha gente, y durante mucho tiempo.

Para conseguir un proyecto de software exitoso:

- ◆ se debe comprender el ámbito del trabajo realizar
- ◆ Riesgos
- ◆ Recursos necesarios
- ◆ Tareas a llevar a cabo
- ◆ Esfuerzo (coste) a consumir
- ◆ Plan a seguir

Todo esto es complejo y debe ser gestionado:

- ◆ Planificarlo
- ◆ Realizar la supervisión
- ◆ Realizar el control de personal, del proceso y los eventos que ocurren.

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

◆ PLANIFICACION DE PROYECTOS

Se trata de estimaciones que hacen los gestores de software, en tiempo limitado , sobre:

- ◆ Recursos
- ◆ Coste
- ◆ Planificación temporal

Estas estimaciones deberían actualizarse regularmente a medida que avanza el proyecto

Para hacerlas se utiliza:

1. Información solicitada a los clientes y a los ingenieros de software
2. Los datos de las métricas de software obtenidos de proyectos anteriores

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- Principal objetivo de la planificación: configuración del calendario del proyecto (también llamado programa de tiempos)
se trata de una representación gráfica de todas las actividades del proyecto necesarias. Esto permite al jefe de proyecto coordinar de una forma efectiva al equipo de desarrollo
A partir de esta planificación, el jefe de proyecto realiza el plan de proyecto: documento que describe los trabajos que se va a realizar y la forma en que el jefe de proyecto va a dirigir su desarrollo

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ Debe cumplir el programa de tiempos (realizado por el gestor del software):
 - ◆ Comprensible para todos
 - ◆ Suficientemente detallado
 - ◆ Capaz de señalar las actividades críticas
 - ◆ Flexible, fácilmente modificable
 - ◆ Basado en estimaciones de tiempo fidedignas Ajustable los recursos disponibles
- ◆ El plan del proyecto

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ El plan del proyecto llevado a cabo por el jefe de proyecto, debe:
 - ◆ definir un conjunto de tareas coordinadas en el tiempo.
 - ◆ Debe definir los recursos necesarios para cumplir los objetivos marcados a cada tarea con el propósito de satisfacer los compromisos adquiridos en desarrollos internos de la empresa

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

◆ CONTROL DE PROYECTOS

- ◆ Después de la planificación del proyecto y comenzada la ejecución del mismo, es importante realizar el control del mismo
- ◆ Se trata de vigilar todas las actividades de desarrollo del sistema
- ◆ Un adecuado control hace posible evitar desviaciones en costes y plazos
- ◆ El jefe de proyecto debe estar muy pendiente, porque cualquier error hay que subsanarlo lo antes posible para solucionarlo sin pérdida de tiempo.

◆ EJECUCION DE PROYECTOS

- ◆ etapa de desarrollo en si
- ◆ Responsabilidad del jefe de proyecto
- ◆ Enfasis a la comunicación para resolver los posibles errores lo antes posible
- ◆ Se deberán organizar regularmente reuniones para administrar el equipo del proyecto, es decir discutir regularmente el progreso del proyecto y determinar las prioridades para las siguientes semanas

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

◆ **HERRAMIENTAS DE USO COMÚN PARA LA GESTION DE PROYECTOS**

- ◆ **EXISTEN HERAMIENTAS DE PAGO:** Microsoft Project (para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al proceso, administrar presupuesto y analizar cargas de trabajo.

◆ **TAMBIEN EXISTEN SOFTWARE DE ACCESO LIBRE**

- ◆ **Colabtiva.** Permite importar desde Basecamp e incluye funciones similares como:
 - ◆ la gestión de diferentes proyectos,
 - ◆ los milsetones
 - ◆ y las listas de tareas.

Pero además:

- ◆ mide el tiempo dedicado a las tareas,
- ◆ emite informes
- ◆ y cuenta con varios plugin para extender sus funciones

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ **Project HQ:**

- ◆ su base de datos es totalmente independiente.
- ◆ Gestiona distintas compañías, miembros.
- ◆ Y cuenta con milestones y listas de tareas
- ◆ Es configurable visualmente usando CSS

- ◆ **Gant PV**

- ◆ gratuito ,
- ◆ sencillo .
- ◆ Se basa en diagramas de Gant para planificaciones de proyectos, y seguimiento de tareas.
- ◆ Disponible para mac, Windods y linux

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ **HERRAMIENTAS DE USO COMÚN PARA LA
GESTION DE PROYECTOS**
- ◆ TAMBIEN EXISTEN SOFTWARE DE ACCESO
LIBRE
 - ◆ Clocking IT :
 - ◆ cuenta con diagramas de Gant interactivos,
 - ◆ contador de tiempo,
 - ◆ varias formas de comunicación,
 - ◆ seguimiento e indexación de los cambios y unos muy buenos informes de avance

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ **HERRAMIENTAS DE USO COMÚN PARA LA GESTION DE PROYECTOS**
- ◆ TAMBIEN EXISTEN SOFTWARE DE ACCESO LIBRE
 - ◆ Team Work
 - ◆ Online.
 - ◆ Permite hacer un seguimiento de distintos proyectos y equipos de trabajo.
 - ◆ Dispone de una versión optimizada para acceder desde móviles.
 - ◆ Disponible para Mac, Windows y Linux
 - ◆

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ **HERRAMIENTAS DE USO COMÚN PARA LA GESTION DE PROYECTOS**
- ◆ TAMBIEN EXISTEN SOFTWARE DE ACCESO LIBRE
 - ◆ iceScrum:
 - ◆ Tiene la misma interfaz para todos los roles
 - ◆ Incluye:
 - ◆ Historial de usuario
 - ◆ Asuntos, problemas
 - ◆ Chat en línea, indicadores de producto

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ **HERRAMIENTAS DE USO COMÚN PARA LA GESTION DE PROYECTOS**
- ◆ TAMBIEN EXISTEN SOFTWARE DE ACCESO LIBRE
 - ◆ Archivo:
 - ◆ Incluye: Calendarios, estadísticas, plantillas y notas
 - ◆ No hay tarifas de licencia o limitaciones para su uso
 - ◆ DotProject.net
 - ◆ Sostenida por voluntarios y usuarios
 - ◆ Permite la gestión para varios clientes
 - ◆ Con herramientas para gestión de tareas, agendas y comunicaciones

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ **HERRAMIENTAS DE USO COMÚN PARA LA GESTION DE PROYECTOS**
- ◆ TAMBIEN EXISTEN SOFTWARE DE ACCESO LIBRE
 - ◆ GranttProject
 - ◆ Disponible para Mac, Windows y Linux
 - ◆ Totalmente gratuito
 - ◆ Incluye diagramas de Grant
 - ◆ Permite exportar los diagramas como imágenes, mientras genera informes en PDF y HTML
 - ◆ Permite interactuar con Project

TEMA1:EL PROCESO DE DESARROLLO DE SOFTWARE

GESTION DE PROYECTOS

- ◆ **HERRAMIENTAS DE USO COMÚN PARA LA GESTION DE PROYECTOS**
- ◆ TAMBIEN EXISTEN SOFTWARE DE ACCESO LIBRE
 - ◆ TaskJungle
 - ◆ Gestor de proyectos realmente potente y superior a otros
 - ◆ Cubre todos los aspectos de desarrollo de un proyecto desde la primera idea hasta su fin
 - ◆ Ayuda a medir
 - ◆ su campo de alcance
 - ◆ La asignación de recursos
 - ◆ Esquemas de costos y ganancias
 - ◆ Riesgo y gestión de las comunicaciones

preguntas

- ◆ Qué modelo de ciclo elegirías, si se trata d en proyecto corto y además el cliente deja claro al principio todos los requisitos
 1. Incremental
 2. RAD
 3. Cascada
- ◆ Cual de estas afirmaciones es falsa sobre las técnicas para realizar la recopilación de requerimientos :
 - En las entrevistas es importante utilizar preguntas abiertas, porque la predisposición y experiencias de la persona entrevistada influirá en la obtención de resultados
 - Análisis de documentos definen la razón de ser del proyecto
 - Tormenta de ideas técnica eficaz porque las ideas más creativas y efectivas son , a menudo , el resultado de la combinación de ideas aparentemente inconexas.
 - Nunca se debe utilizar preguntas abiertas en las entrevistas

preguntas

◆ Cual de estas afirmaciones es falsa:

- ◆ El requisito es aquello que necesita un usuario para conseguir un objetivo determinado a la hora de desarrollar un software.
- ◆ el diseño de sistemas es el arte de definir la arquitectura de hardware y software , componentes, módulos y datos de un sistema de computo para satisfacer ciertos requerimientos. Es la etapa posterior al análisis de sistemas.
- ◆ La implementación es la etapa en la que se comienza a codificar el diseño realizado en la etapa anterior.
- ◆ La Validación es el conjunto de actividades que aseguran que el software implementa correctamente una función específica, mientras que la verificación n: se refiere a un conjunto diferente de actividades que aseguran que el software construido, se ajusta a los requisitos del cliente.
Es esta, porque es al revés. Validación/verificación

◆ De que forma se puede probar un software:

- ◆ Conociendo la función específica para la que fue diseñado el producto, se van llevando a cabo pruebas que demuestren que cada función es completamente operativa al tiempo que se busca errores en cada función.
- ◆ Conociendo el funcionamiento del producto se pueden realizar pruebas que aseguren que todas las piezas encajan , o sea que la operación interna se ajusta a los requisitos
- ◆ Las dos a la vez

preguntas

- ◆ Que prueba se encargaría de responder una pregunta como ¿ a qué potencia puedo ponerlo a funcionar antes de que falle?
 - ◆ **Pruebas de recuperación.**
 - ◆ **Pruebas de seguridad.**
 - ◆ **Pruebas de resistencia**
 - ◆ **Pruebas de rendimiento**

preguntas

- ◆ Cual es de estas respuestas la incorrecta
 - ◆ Los compiladores son programas que traducen el fichero de código fuente a lenguaje máquina
 - ◆ Los enlazadores son programas que enlazan varios ficheros objeto en lenguaje binario para crear un único fichero , el ejecutable del programa
 - ◆ Ninguna es incorrecta

preguntas

- ◆ Cual de estos objetos no corresponde a la planificación de proyecto
 - ◆ Medir el tiempo dedicado a las tareas,
 - ◆ Emitir informes
 - ◆ extender sus funciones mediante plug-ins
 - ◆ Ejecutar el proyecto

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ESTRUCTURADA Y TIPOS ABSTRACTOS DE DATOS (ADT)

2.1 PROGRAMACION ESTRUCTURADA Y TIPOS ABSTRACTOS DE DATOS (ADT)

la programación estructurada (También llamada modular):consisten en el uso de abstracción de procedimientos.

La idea es meter un conjunto de instrucciones en un módulo, que tiene una interfaz claramente definida. Mediante la abstracción de procedimientos el programador separa claramente el **cómo** del **qué**: una subrutina contiene la lógica necesaria para ejecutar una tarea, y sus parámetros definen los objetos sobre los que trabaja.

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ESTRUCTURADA Y TIPOS ABSTRACTOS DE DATOS (ADT)

2.1 PROGRAMACION ESTRUCTURADA Y TIPOS ABSTRACTOS DE DATOS (ADT)

la programación estructurada Provee tres conjuntos de estructuras de control, Las cuales permiten adaptar el código fuente en cualquier situación:

- ◆ **Instrucciones secuenciales**

aquellas que se ejecutan de forma secuencial o en orden, una tras otra .
Dentro de estas encontramos las instrucciones de asignación, operaciones sobre variables....

- ◆ **instrucciones de selección**

También conocidas como condicionales, permiten tomar decisiones dentro del código en base al contexto de ejecución del programa

- ◆ **Instrucciones de iteración**

llamadas instrucciones de repetición o bucles, permiten ejecutar un bloque de código de forma repetitiva, hasta que se alcanza una determinada condición o estado en el programa

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ESTRUCTURADA Y TIPOS ABSTRACTOS DE DATOS (ADT)

Todo ADT se compone de dos interfaz:

- ◆ Una interfaz pública en la que se hace visible el resto de las estructuras de datos de un programa, sus datos y operaciones, es decir su comportamiento
- ◆ Una interfaz privada (parte oculta) en la que se recoge la implementación de las operaciones así con la inicialización de los datos del propio ADT

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ESTRUCTURADA Y TIPOS ABSTRACTOS DE DATOS (ADT)

Esta forma de estructurar los datos permite a los programadores:

- ◆ Encapsular(empaquetar y organizar) , los datos y operación de las entidades de un programa, dando lugar a unidades de información independientes y reutilizables
- ◆ Ocultar la información irrelevante al resto de estructura de datos de un programa, lo que permite dejar a la vista solo aquellos datos y operaciones que pueden ser utilizadas

Conclusión esta forma de estructurar los programas supuso un gran avance, porque permite crear estructura de de datos reutilizables para las implementaciones de otros programas, resultando un gran avance.

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS. CLASES Y OBJETOS

- La Programación Orientada a Objetos (POO) nos enseña un método - probado y estudiado- el cual se basa en las interacciones de objetos para resolver las necesidades de un sistema informático.

- Clases**

Las clases son los modelos sobre los cuáles se construirán nuestros objetos
EJEMPLO: ANTENA, PELO, OJO

Agregación de clases

En los POO las clases no constan solo de atributos simples, sino que suelen componerse de clases que a su vez se componen de otras clases hijas, dando lugar a una agregación de clases.

Ej. clase coche puede almacenar al propietario a través de un objeto PropietarioCoche y el seguro a través de una clase SeguroCoche

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS. CLASES Y OBJETOS

- ◆ **Propiedades.** son las características intrínsecas del objeto.
EJEMPLO

- ◆ ANTENA:
 - ◆ COLOR
 - ◆ LONGITUD
- ◆ PELO
 - ◆ COLOR
 - ◆ TEXTURA
- ◆ OJO
 - ◆ FORMA
 - ◆ COLOR
 - ◆ TAMAÑO

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS. CLASES Y OBJETOS

- ◆ **Gestión de excepciones: tratamiento de errores**
- ◆ Desde los inicios de los lenguajes de programación, la gestión de errores ha sido uno de los asuntos más difíciles.
- ◆ Es tan complicado diseñar un buen esquema de gestión de errores, que muchos lenguajes simplemente lo ignoran,
El problema más importante de la mayoría de los esquemas de gestión de errores es que dependen de que el programador se preocupe en seguir un convenio que no está forzado por el lenguaje. Si los programadores no se preocupan, cosa que ocurre cuando se tiene prisa, esos esquemas se olvidan fácilmente.
- ◆ Merece la pena tener en cuenta que la gestión de excepciones no es una característica orientada a objetos, aunque en lenguajes orientados a objetos las excepciones normalmente se representan con objetos. La gestión de excepciones existía antes que los lenguajes orientados a objetos.

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS. CLASES Y OBJETOS

Objeto

- Las clases por sí mismas, no son más que modelos que nos servirán para crear objetos en concreto.

Podemos decir que una clase, es el razonamiento abstracto de un objeto, mientras que el objeto, es su materialización.

A la acción de crear objetos, se la denomina *instanciar una clase* y dicha instancia, consiste en asignar la clase, como valor a una variable:

OBJETO OJO X: VERDE, GRANDE, FEO,....

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS. CLASES Y OBJETOS

◆ Objeto: creación de objetos

La forma de instancia un objeto en un lenguaje orientado a objetos se realiza generalmente utilizando un operador o palabra reservada del mismo , como es el caso de NEW.

Crea un objeto (instancia) de una clase declarada dentro de un programa, y cuya referencia (dirección de memoria), suele almacenarse en una variable con el mismo tipo que define la clase

ej. Coche miCoche= **new** coche

las clases pueden implementar unos métodos de instancias especiales, denominados **constructores**

los constructores pueden inicializar los objetos a instanciar con datos.

Los constructores no devuelven ningún valor

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS. CLASES Y OBJETOS

💧 Objeto: paso de mensajes

Es como se llama a la comunicación básica entre objetos es el mecanismo que permite a un objeto A llamar al método de un objeto B al cual se le puede pasar o no información adicional a través de parámetros

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS. CLASES Y OBJETOS

Objeto: Visibilidad de atributos y métodos . Modificación de acceso

diferencias entre los diferentes niveles de visibilidad:

- ◆ **Public:** El atributo o método que podrá ser leído o llamado en cualquier parte del código, sea desde la misma clase o desde fuera de ella, esto significa que también podrá ser alterado sin restricciones,
- ◆ **Private:** El atributo o método que solo puede ser leído o alterado desde el interior de la clase. Al intentar acceder desde fuera genera un error
- ◆ **Protected:** Los atributos o métodos protected son similares a los private, solo se puede acceder a ellos desde el interior de la clase, pero con una diferencia muy útil, las clases heredadas de esta también podrán leer y alterar estos atributos.

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS. CLASES Y OBJETOS

◆ Objeto: Referencias a objetos

Durante la declaración de las clases que conforman un programa orientado a objetos , en la implementación de constructores y métodos de instancia, puede hacerse uso de la palabra reservada **THIS** para hacer referencia al propio objeto instanciado de una determinada clase declarada

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS. CLASES Y OBJETOS

◆ Objeto: **persistencia de objetos**

Conjunto de técnicas que permiten salvaguardar la información de un programa en ejecución

es habitual que cuando finaliza la ejecución de un programa, se destruya todos los objetos que encuentran en memoria, dando lugar a la pérdida de datos e información que contenían

Para poder guardar y reutilizar dichos datos es preciso utilizar técnicas de **serialización**, los cuales permiten almacenar y recuperar los datos de un programa en memoria volátil, ya sea mediante el uso de ficheros o por medio de una base de datos.

TEMA2:

LA ORIENTACION A OBJETOS

PROGRAMACION ORIENTADA A OBJETOS.

CLASES Y OBJETOS

◆ **Objeto: destrucción de objetos**

Al finalizar la ejecución de un programa orientado a objetos, es habitual, que antes de hacerlo se lleve a cabo de forma manual o automática, la destrucción de todos los objetos instanciados por el programa que los creo. Se trata de liberar la memoria principal asociada a los objetos creados para que esta pueda ser reutilizada por el resto de programas en ejecución del computador.

Puede ser un

- ◆ proceso automático.ej. Java que dispone de el mecanismo llamado recolector de basura, encargado de detectar cuando un objeto de un programa en ejecución deja de ser utilizado o referenciado por el mismo y a su vez , de forma automática eliminado de memoria
- ◆ o manual . Ej. C++ que puede utilizar el método finalice() sobre el objeto a eliminar.,

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

- ◆ Concepto fundamental de la programación orientada a objetos
- ◆ Permite:
 - ◆ establecer jerarquías o relaciones conceptuales padre-hijo entre clases de objetos
 - ◆ Crear clasificaciones de clases por tipo, creando relaciones entre las distintas clases de objetos que componen un programa

Las clases hijas siempre son conceptos mas especializados que las clases padre. Por eso denominamos al proceso de creación de clases hijos como **especialización**

Por el contrario , cuando pensaos en crear una clase padre a partir de clases hijas, a este proceso se le denomina como **generalización**

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

◆ HERENCIA MULTIPLE

La herencia múltiple hacer referencia a la situación en la que una clase hija puede definir como superclases dos o más clases padre pudiendo heredar el comportamiento y características de estas.

Los problemas surgen cuando las clases padre comparten comportamientos parecidos entres si, lo que da lugar a inconsistencias y ambigüedades en las clases hija, dado que estas a la hora de ejecutar su comportamiento , son incapaces de determinar qué comportamiento de sus clases padre llevar a cabo

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

- ◆ CLASE ABSTRACTA: En ocasiones un conjunto de clases hijas a veces comparten un comportamiento (método común, el cual no está definido en la clase padre de la cual heredan.
- ◆ En estas ocasiones la POO utiliza un mecanismo el cual permite definir a través de una clase padre abstracta el comportamiento común a las subclases, pero no así la implementación que será definida de forma particular por cada subclase
- ◆ Características de la clase abstracta:
 - ◆ Poseen un comportamiento sin implementar: métodos abstractos sin código
 - ◆ Una clase abstracta no se puede instanciar, es decir crear objetos a partir de ella

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

♦ MODULARIDAD. LIBRERIAS DE CLASE

Los lenguajes orientados a objetos , tienen mecanismos o herramientas que permiten agrupar y empaquetar las clases implementadas en paquetes o librerías de clases , con una temática o funcionalidad similar.

♦ GENERICIDAD Y SOBRECARGA DE MÉTODOS

Genericidad:

Mecanismo que tiene la POO a través de la cual se puede parametrizar clases y colecciones de datos con tipos genéricos (sin una clase definida), lo que posibilita crear código fuente reutilizable durante la implementación de clases y estructuras de datos.

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

Sobrecarga de métodos:

Mecanismo orientado a la reutilización de código de la POO.

Permite al programador reutilizar el nombre de los métodos de las clases ya implementados , y realizar nuevas implementaciones que utilicen distinto número y tipo de argumento.

Para poder ser llevada a cabo la sobrecarga, debe cumplir:

- ◆ Dos métodos sobrecargados deben tener distintos argumentos
- ◆ Pueden presentar distinto tipo de valor de retorno
- ◆ Pueden tener distinto modificador de acceso
- ◆ Pueden declarar distintas excepciones
- ◆ Por ultimo, pueden encontrarse en la misma clase o en una subclase

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

Comparación entre genericidad y sobrecarga:

- ◆ Ambas proveen mecanismo de programación que mejoran la abstracción de código y a su vez favorecen la reutilización del mismo
- ◆ Permiten implementar código con estructura genérica , válido para una amplia diversidad de tipos de datos .
- ◆ A su vez estas técnicas reutilizan también el comportamiento , como son los nombres de los métodos, que realizan una misma función pero con los mismos tipos de datos

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

DESARROLLO ORIENTADO A OBJETOS

Lenguajes de desarrollo orientado a objetos de uso común.

◆ Java

- ◆ Lenguaje de programación multiplataforma e propósito general
- ◆ Comercializado por primera vez en 1995 por Sun Microsystems
- ◆ Actualmente muy utilizados en el desarrollo de varias aplicaciones y sitios web
- ◆ Es rápido seguro y fiable
- ◆ Puede ser utilizado en:
 - ◆ Pequeños sistemas embebidos
 - ◆ En grandes centros de datos y cómputo
 - ◆ En consolas
 - ◆ Teléfonos móviles
 - ◆ Portales de internet
- ◆ **Filosofía WORE:** " escribe una vez el código y ejecutado en cualquier plataforma". Esto se debe a que Java no es específico de una plataforma concreta, sino que su código puede exportarse a cualquiera de ellas puesto que utiliza una máquina virtual que se encarga de convertir de forma automática el código byte generado por el compilador de Java , al lenguaje máquina específico de cada plataforma, que se caracteriza por estar orientado a objetos y ser independiente de cualquier plataforma y portable

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

DESARROLLO ORIENTADO A OBJETOS

Lenguajes de desarrollo orientado a objetos de uso común.

💧 PHP

- 💧 Uno de los lenguajes de uso general más extendidos actualmente en el mundo web e internet, debido a que cuenta con una gran comunidad de desarrolladores al tratarse de un lenguaje gratuito y de código abierto, además de ser soportado por múltiples plataformas
- 💧 **Fácil de aprender, flexible, potente y de alto rendimiento, lo que le lleva a convertirse en el lenguaje número uno en el desarrollo de sistemas web**

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

DESARROLLO ORIENTADO A OBJETOS

Lenguajes de desarrollo orientado a objetos de uso común.

◆ Python

- ◆ Lenguaje de scripting
- ◆ Independiente de cualquier plataforma y orientado a objetos
- ◆ Permite realizar cualquier tipo de programas: aplicaciones Windows, servidores de red, e incluso páginas web
- ◆ Lenguaje interpretado por lo que no requiere compilar el código fuente para poder ejecutarlo
- ◆ Últimamente muy popular gracias a:
 - ◆ la cantidad de librerías , tipos de datos y funciones incorporadas en el propio lenguaje
 - ◆ A la sencillez y velocidad con la que se crean los programas
 - ◆ A ser gratuito

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

DESARROLLO ORIENTADO A OBJETOS

Herramientas de desarrollo
entre todas las soluciones software para el desarrollo de aplicaciones, podemos destacar:

- ◆ Eclipse
 - ◆ Entorno de desarrollo integrado IDE, de código abierto
 - ◆ Desarrollado en Java
 - ◆ Plataforma de desarrollo libre y uso gratuita, de código abierto,
 - ◆ En constante evolución dado que cuenta con una amplia comunidad de desarrolladores
 - ◆ Ofrece numerosos útiles de soporte y ayuda a la programación para diversos lenguajes tanto de uso general como para lenguajes de marcado:
 - ◆ Gestión de servidores web, bases de datos y gestión de código fuente
 - ◆ Plug-ins, destinadas a agilizar y facilitar las labores de desarrollo y codificación de programas

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

DESARROLLO ORIENTADO A OBJETOS

- ◆ NetBeans
 - ◆ Lenguaje de programación Java
 - ◆ Surgió de la mano de Sun Microsystems.
Esta plataforma gratuita y de código abierto no presenta restricciones de uso en su licencia, lo cual permite a los desarrolladores más avanzados poder extender dicha plataforma a sus necesidades
 - ◆ Estructura interna basada en módulos, los cuales ofrecen extensibilidad de la plataforma a innumerables funcionalidades

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

LENGUAJES DE MODELIZACION EN EL DESARROLLO ORIENTADO A OBJETOS

Son lenguajes utilizados en el entorno empresarial, antes de el desarrollo, que permiten abordar las fases de análisis y diseño de los sistemas de información.

Consiste en un lenguaje gráfico que permite a través de una serie de diagramas especificar, construir y documentar un sistema de información de principio a fin.

TEMA2:

LA ORIENTACION A OBJETOS

HERENCIA

LENGUAJES DE MODELIZACION EN EL DESARROLLO ORIENTADO A OBJETOS

Dicho diagrama se compone de tres gráficos:

Diagramas de Casos de Uso interpreta como interactúa un usuario o actor con el sistema de información en desarrollo a través de las operaciones o casos de uso

relación de inclusión (<incluye>) requieren de realización previa de otro caso de uso antes de realizar su propia operación

Relación de Extensión(<extends>) permiten especificar casos de uso que extienden la funcionalidad de un caso de uso existente

Diagrama de clases para representar y documentar de forma gráfica las clases con sus atributos y métodos

relación de herencia Utilizado en casos en los que se requiere que una clase hija herede los atributos y métodos de la clase padre

relación de asociación relación entre dos clases que colaboran entre si

relación de composición y agregación permite establecer relaciones de posesión y pertenencia entre clases.

Diagrama de Interacción Diagrama que se utiliza para documentar como interaccionan los objetos de una clase con los de otra, a partir de un evento generado

TEMA3:

ARQUITECTURAS WEB

- La creación de las páginas web dinámicas permite separación entre:
 - Modelo físico . Arquitectura multinivel . Este tipo de arquitectura propone distribuir los elementos de hardware en los que se ejecutaran los procesos. Cada elemento una capa diferente
 - Debemos resaltar la diferencia entre capa y nivel:
 - Nivel: forma física en la que se organiza una aplicación
 - Capa: distintas partes en que se divide una aplicación

TEMA3:ARQUITECTURAS WEB

Modelos físicos de separación: Arquitecturas Multinivel

- ◆ Teniendo en cuenta que las arquitecturas pueden tener los siguientes niveles:
 - ◆ Ejemplo Dos niveles: cliente-servidor
 - ◆ Ejemplo tres niveles: cliente (dispositivos móviles, servidor web, servidor de BD)
 - ◆ Ejemplo de arquitectura estándar para la construcción de páginas web (4 niveles) cliente(portátiles), Servidor web, Servidor de aplicaciones, Servidor de bases de datos
 - ◆ Ejemplo 2 de arquitectura de 4 niveles: portátiles, servidor de aplicaciones, servidor de Bases de Datos, Servidor de Streeming

TEMA3:ARQUITECTURAS WEB

Modelo de separación lógicos

Uno de los mecanismos más usados a la hora de construir páginas web en el uso de las arquitecturas e capas.

Estas arquitectura dividen el desarrollo de una aplicación en varios niveles lógicos.

Ventajas:

- ◆ Desarrollos paralelos en cada capa
- ◆ Aplicaciones mas robustas debido al encapsulamiento
- ◆ Mantenimiento y soporte más sencillo. Es más fácil cambiar un componente.
- ◆ Mayor flexibilidad . Podemos añadir nuevos módulos para dotar al sistema de nueva funcionalidad
- ◆ Alta escalabilidad. Podeos manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware

Las arquitecturas en capas son una de las formas de implementación de un patrón arquitectónico denominado **Modelo-Vista-Controlador (MVC)**

ARQUITECTURAS WEB

Modelo de separación lógicos

- ◆ **El esquema modelo-vista-controlador (MVC)**

Este esquema busca separar una aplicación en tres componentes principales:

- ◆ **El modelo:** partes de la aplicación que implementan la lógica de la aplicación para un dominio específico. Se trata de la representación de la información con la cual se opera. Normalmente los modelos devuelven y almacenan el estado del modelo en una base de datos
- ◆ **La vista.** Son los componentes que despliegan la interfaz de usuario. . Se trata del conjunto de páginas web que muestran y recogen la información de usuario
- ◆ **El controlador.** Se trata de el componente que maneja la interacción con el usuario, trabajan con el modelo y seleccionan cual es la vista correcta a desplegar para mostrar la información

ARQUITECTURAS WEB

Modelo de separación lógicos

- ◆ El patrón MVC nos ayuda a crear aplicaciones que separan los diferentes aspectos de la aplicación, la lógica de entradas, la lógica de negocio y la lógica a de interfaz, generando un acoplamiento pequeño entre cada una de estas partes.
- ◆ Por otro lado el patrón especifica que cada una de estas lógicas debe corresponderse con un elemento

ARQUITECTURAS WEB

Modelo de separación lógicos

- ◆ Conclusión:

- ◆ La lógica de interfaz pertenecerá a **la vista**
- ◆ La lógica de entradas pertenecerá **al controlador**
- ◆ La lógica de negocio pertenecerá **al modelo**

ARQUITECTURAS WEB

Modelo de separación lógicos

- ◆ **La arquitectura de tres capas**

- ◆ Es una de las más utilizadas para el desarrollo de sitios web. Esta arquitectura divide la creación de una aplicación en tres niveles

- ◆ Capas de presentación
- ◆ Capas de negocio
- ◆ Capa de persistencia

El modelo de tres capas está destinado a ayudar a construir componentes de software a partir de la división de los niveles lógicos

ARQUITECTURAS WEB

Modelo de separación lógicos

◆ La arquitectura de tres capas

◆ Capa de presentación

- ◆ También llamada de interfaz gráfica .La capa más cercana al usuario
- ◆ Se comunica con la capa de negocio y suele estar dividida entre los niveles de cliente y servidor
- ◆ Tareas:
- ◆ Proveer de la página web
- ◆ Garantizar que dichas páginas puedan ser visualizadas tanto en un ordenador como en un dispositivo smartphone, gestionar que todos los enlaces del sitio web sean válidos, garantizar que las páginas web tengan siempre la información actualizada....

ARQUITECTURAS WEB

Modelo de separación lógicos

◆ La arquitectura de tres capas

◆ Capa de negocio

- ◆ También denominada capa de lógica, capa de aplicación, capa media o capa de lógica de negocio.
- ◆ Es a capa que gestiona las funcionalidades del sistema o aplicación web
- ◆ Habitualmente esta capa recibe las peticiones del usuario y desde ella se envían las respuestas apropiadas tras el procesamiento de la información proporcionada por el cliente.
- ◆ Es en esta capa donde se establecen todas las reglas que deben cumplirse para que las funciones de la aplicación web sean ejecutadas correctamente.
- ◆ Tareas de los componentes de esta capa:
 - ◆ Coordinar la aplicación web
 - ◆ Procesar los comandos
 - ◆ Realizar la toma de decisiones
 - ◆ Realizar los cálculos respectivos
 - ◆ Mover y procesar los datos entre sus capas adyacentes

ARQUITECTURAS WEB

Modelo de separación lógicos

💧 **La arquitectura de tres capas**

- 💧 Capa de persistencia: Es la encargada del acceso a los datos

aquí se encuentra toda la codificación necesario tanto para el acceso a los datos como para el manejo y almacenamiento de los mismos.

Normalmente está formada por un conjunto de componente que se encargad del acceso a los datos , los cuales se ejecutan en los servidores de aplicaciones y por uno o más gestores de bases de datos

ARQUITECTURAS WEB

Modelo de separación lógicos

- ◆ **Arquitecturas multicapa**

- ◆ Una arquitectura multicapas es un conjunto ordenado de subsistemas, cada uno de cuales está constituido en términos de los que tiene por debajo y proporciona la base de la implementación de aquellos que están por encima de él.
- ◆ Los objetos de cada capa suelen ser independientes, aunque suelen haber dependencias entre objetos de distintas capas. Existe una relación cliente /servidor entre las capas inferiores, que son las que proporcionan los servicios, y las capas superiores, los usuarios de estos servicios.
- ◆ Una arquitectura multicapa particional todo el sistema en distintas unidades funcionales: cliente, presentación, lógica-de-negocio, integración, y sistema de información empresarial (EIS). Esto asegura una división clara de responsabilidades y hace que el sistema sea más mantenerle y extensible. Los sistemas con tres o más capas se han probado como más escalables y flexibles que un sistema cliente-servidor, en el que no existe la capa central de lógica de negocios. La capa de presentación expone los servicios de la capa de lógica de negocio a los usuarios. Sabe cómo procesar una petición de cliente, cómo interactuar con la capa de lógica de negocio, y cómo seleccionar la siguiente vista a mostrar. La capa de la lógica de negocio contiene los objetos y servicios de negocio de la aplicación. Recibe peticiones de la capa de presentación, procesa la lógica de negocio basada en las peticiones, y media en los accesos a los recursos de la capa EIS. Los componentes de la capa de lógica de negocio se benefician de la mayoría de los servicios a nivel de sistema como el control de seguridad, de transacciones y de recursos. La capa del cliente es donde se consumen y presentan los modelos de datos. Para una aplicación Web, la capa cliente normalmente es un navegador web. Los clientes pequeños basados en navegador no contienen lógica de presentación; se trata en la capa

ARQUITECTURAS WEB

Modelo de separación lógicos

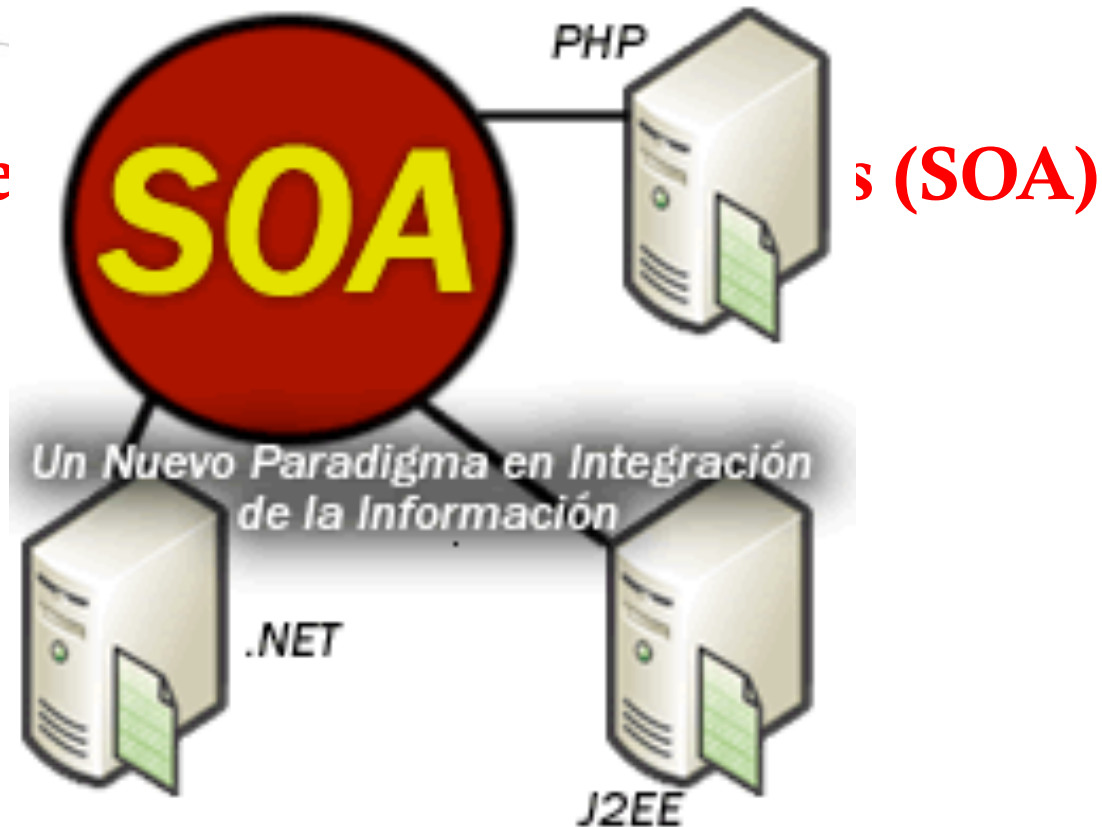
💧 **Arquitectura orientada a servicios (SOA)**

- 💧 SOA permite emular el comportamiento de los negocios en el mundo real
- 💧 En una Arquitectura Orientada a Servicios , los usuarios finales , mediante la utilización de un hardware software liviano como un navegador web pueden acceder a lo que se denomina el nivel de clientes o aplicaciones, que básicamente está constituido por la capa de presentación donde consumen los servicios publicados por una organización.
- 💧 Este tipo de servicios son , en general sitios o portales en la web.
- 💧 Las aplicaciones de otras organizaciones también pueden utilizar los servicios por una organización concreta; esta acción requiere de acuerdos comerciales y credenciales para autenticar y autorizar a quienes consumen los servicios

ARQUITECTURAS WEB

Modelo de separación lógicos

● **Arquite**



ARQUITECTURAS WEB

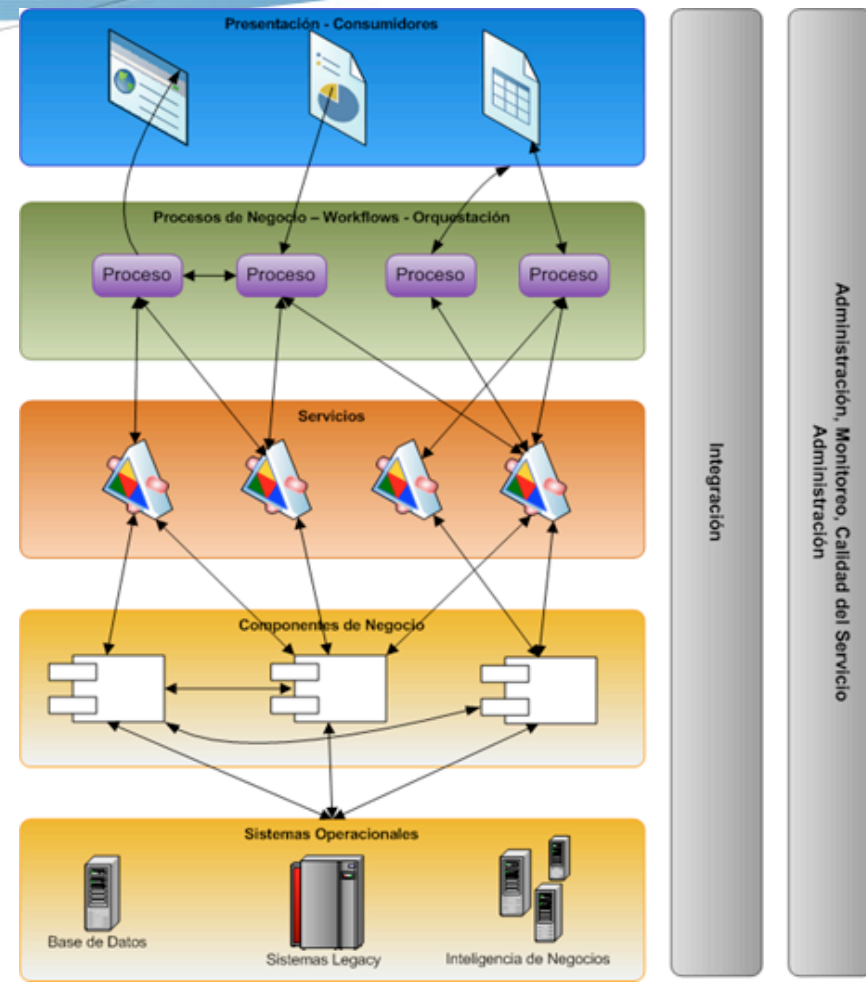
Modelo de separación lógicos

🔹 Arquitectura orientada a servicios (SOA)

El proceso de orientación a servicios puede dividirse en cuatro capas:

La cuarta es una capa de análisis en la que se determinan los procesos de negocios. Es decir la lógica del negocio se refiere a la realidad de la organización, a los procesos que guían la manera en la que se hacen las cosas en el mundo real, a los procesos que especifican los requisitos y proponen soluciones de negocio. Además en estas capas se encargan de crear las estrategias de integración y reutilización de los servicios

Finalmente la capa de los sistemas operacionales e implementada por analistas del sistema o desarrolladores de servicios. Es en esta capa donde se desarrollan los servicios web.



ARQUITECTURAS WEB

Herramientas de desarrollo orientadas a servidor de aplicaciones web

Las aplicaciones web están formadas por un conjunto de páginas HTML, scripts de lenguajes web , programas escritos en diferentes lenguajes, bases de datos y documentos de diferentes formatos y pueden residir en varias ubicaciones o sitios web, es decir, en diferentes servidores.

A la hora de seleccionar las herramientas que un desarrollador tiene a su disposición es necesario hacer una primera clasificación de los instrumentos involucrados en función de sus capacidades:

- ◆ **Navegadores** se ejecutan en el entorno del cliente y que permiten visualizar los documentos escritos en lenguaje HTML y capturar las interacciones del usuario
- ◆ **Editores de documentos.** Editores de texto que permiten escribir código HTML directamente, sin ninguna ayuda ni facilidad adicional.
- ◆ **Entornos de programación** Son entornos integrados que nos permiten editar, compilar y ejecutar los programas generados a partir de diferentes lenguajes usado en el desarrollo de aplicaciones web
- ◆ **Herramientas de tratamiento de imágenes.** La mayoría de las páginas web muestran contenido gráfico de una u otra manera, por ello es necesario el uso de este tipo de herramientas para optimizar las imágenes en la web

ARQUITECTURAS WEB

Herramientas de desarrollo orientadas a servidor de aplicaciones web

◆ **Marcadores de texto**

son editores de texto dirigidos al ámbito de la programación, reconocimiento de sintaxis de lenguaje :

- ◆ **Notepad++**. Editor gratuito de código fuente. Soporta varios lenguajes de programación y se ejecuta en windows
- ◆ **Arachnophilia**. Robusto editor de texto , ideal para programar sitios web en diferentes lenguajes: HTML, JS, C++, CGL, Perl, Java....
También incluye un cliente inteligente de FTP para subir automáticamente los archivos modificados
- ◆ **UltraEdit**: Completo editor de texto para programación. Soporta múltiples formatos con colores configurados para cada lenguaje. Es uno de los múltiples editores de pago que podemos encontrar en el mercado

◆ **Herramientas genéricas**

◆ Tienen más funcionalidades además del reconocimiento de sintaxis de lenguaje.
Añaden capacidades como la sugerencia de estructuras o funciones predeterminadas o la posibilidad de validar la corrección del código escrito

- ◆ **FrontPage**: Es una herramienta de construcción y edición de páginas web para Windows
- ◆ **Eclipse** Entorno de programación gratuito escrito en Java que permite desarrollar aplicaciones en varios lenguajes . Una de sus principales características es que permite la extensión de sus funcionalidades mediante la instalación de múltiples módulos para diferentes propósitos específicos
- ◆ **Dreamweaver** Es un editor de código HTML profesional de pago para el diseño visual y la administración de sitios y páginas web también incluye HTML, CSS y JS

ARQUITECTURAS WEB

Herramientas de desarrollo orientadas a servidor de aplicaciones web

◆ **Herramientas específicas**

- ◆ En muchas ocasiones, la utilización de un lenguaje de programación determinado exige que el desarrollador tenga instalada una plataforma de desarrollo concreta. Tal es el caso de la programación en Java que requiere la instalación de la Máquina Virtual de Java(JVM) o del entorno .Net Framework de Microsoft, necesario para la programación con ASP.Net.

A continuación mostramos una serie de herramientas para una tecnología concreta:

- ◆ Microsoft Visual Studio. Es el entorno e desarrollo más conocido para el diseño de aplicaciones en los sistemas operativos de Microsoft .
Esta destinad al desarrollo de aplicaciones web con lenguajes como Visual Basic o C++ y permite la publicación de la aplicación Web que se está desarrollando directamente desde su interfaz
- ◆ NetBenas IDE. Es una aplicación de código abierto diseñada opera el desarrollo de aplicaciones fácilmente portables entre distintas plataformas haciendo uso de las tecnología Java . Se trata de un entrono de desarrollo gratuito optimizado para el desarrollo de aplicaciones de Java

preguntas

- ◆ Qué modelo de ciclo elegirías, si se trata de un proyecto corto y además el cliente deja claro al principio todos los requisitos
 1. Incremental
 2. RAD
 3. Cascada
- ◆ Cual de estas afirmaciones es falsa sobre las técnicas para realizar la recopilación de requerimientos :
 - En las entrevistas es importante utilizar preguntas abiertas, porque la predisposición y experiencias de la persona entrevistada influirán en la obtención de resultados
 - Análisis de documentos definen la razón de ser del proyecto
 - Tormenta de ideas técnica eficaz porque las ideas más creativas y efectivas son , a menudo , el resultado de la combinación de ideas aparentemente inconexas.
 - Nunca se debe utilizar preguntas abiertas en las entrevistas

preguntas

◆ Cual de estas afirmaciones es falsa:

- ◆ El requisito es aquello que necesita un usuario para conseguir un objetivo determinado a la hora de desarrollar un software.
- ◆ el diseño de sistemas es el arte de definir la arquitectura de hardware y software , componentes, módulos y datos de un sistema de computo para satisfacer ciertos requerimientos. Es la etapa posterior al análisis de sistemas.
- ◆ La implementación es la etapa en la que se comienza a codificar el diseño realizado en la etapa anterior.
- ◆ La Validación es el conjunto de actividades que aseguran que el software implementa correctamente una función específica, mientras que la verificación n: se refiere a un conjunto diferente de actividades que aseguran que el software construido, se ajusta a los requisitos del cliente.
Es esta, porque es al revés. Validación/verificación

◆ De que forma se puede probar un software:

- ◆ Conociendo la función específica para la que fue diseñado el producto, se van llevando a cabo pruebas que demuestren que cada función es completamente operativa al tiempo que se busca errores en cada función.
- ◆ Conociendo el funcionamiento del producto se pueden realizar pruebas que aseguren que todas las piezas encajan , o sea que la operación interna se ajusta a los requisitos
- ◆ Las dos a la vez

preguntas

- ◆ Que prueba se encargaría de responder una pregunta como ¿ a qué potencia puedo ponerlo a funcionar antes de que falle?
 - ◆ **Pruebas de recuperación.**
 - ◆ **Pruebas de seguridad.**
 - ◆ **Pruebas de resistencia**
 - ◆ **Pruebas de rendimiento**

preguntas

- ◆ Cual es de estas respuestas la incorrecta
 - ◆ Los compiladores son programas que traducen el fichero de código fuente a lenguaje máquina
 - ◆ Los enlazadores son programas que enlazan varios ficheros objeto en lenguaje binario para crear un único fichero , el ejecutable del programa
 - ◆ Ninguna es incorrecta

preguntas

- ◆ Cual de estos objetos no corresponde a la planificación de proyecto
 - ◆ Medir el tiempo dedicado a las tareas,
 - ◆ Emitir informes
 - ◆ extender sus funciones mediante plug-ins
 - ◆ Ejecutar el proyecto

Preguntas abiertas

- ◆ Describe los inconveniente de utilizar el modelo de ciclo incremental
Que requiere gran experiencia para definir incrementos
y que puede dar problemas con la arquitectura por no cumplir todos los requisitos
- ◆ Para qué tipo de proyectos es adecuado utilizar el modelo de ciclo de vida del software interactivo?
- ◆ Para proyectos cambiantes
- ◆ y el de cascada?
Para proyectos no cambiantes estables. Proyectos pequeños donde los requisitos están bien entendidos
- ◆ descríbeme algunas técnicas para realizar recopilación de requerimientos pago 13

Preguntas abiertas

- 💧 Qué es el diseño de sistemas? Pág. 16 (pág. 27 del libro)
- 💧 cual de estos no se puede considerar objetivo del Diseño de sistemas? (pagina del libro 28)
- 💧 Separar el modelo de análisis de el de implementación (no es separar precisamente es acercar)
- 💧 definir la interfaz de cada subsistema
- 💧 representación arquitectónica del sistema

Preguntas abiertas

- ◆ Define qué es la implementación pág. 19
- ◆ Define los objetivos de validación y verificación de sistemas. Pág. 21 (página 30)
- ◆ ¿para qué se realiza la prueba de software o testing pág. 24 (
- ◆ Diferencias entre pruebas de software de caja blanca y de caja negra .pág. 27
- ◆ Dime concretamente que se debe comprobar con las pruebas de software. Pág. 34 (libro pág. 33)
- ◆ Enumérame solo cuales son los 4 tipos de pruebas de sistema que existen: (PAGINA 34)de recuperación de seguridad
,de resistencia, de rendimiento
- ◆ hay algún software QUE permita la automatización de pruebas ¿cual ?. Pág. 37 (PAG 37)

Preguntas abiertas

- ◆ Dime cuales son los principios de calidad de software y cual de desarrollo de software
 - ◆ corrección, robustez, eficacia, portabilidad, integridad, facilidad de uso, extensibilidad (calidad de software) PAG 39 (PAG ·36)
 - ◆ rigor y formalidad, separación de intereses, modularidad, abstracción, anticipación del cambio, generalidad, incrementabilidad (PRINCIPIOS BÁSICOS DE DESARROLLO DE SOFTWARE) PAG 19

Preguntas abiertas

- ◆ Dime todo lo que sepas sobre relación entre la calidad del software y métricas pág. 40 (pág. del libro 36)
- ◆ Dime las principales métricas en las fases de ciclo de vida del software. PAG 40
- ◆ Cuéntame que función tiene el calendario en la gestión de proyectos. Pagina 41 libro
- ◆ Háblame de la programación estructurada y el uso de tipos abstractos de datos pagina 64 (pág. 46)
- ◆ háblame de los 2 interfaz de los que se compone la ADT (66) Pág. 47
- ◆ Que son las clases en la poo (68)Pág. 47
- ◆ dime que es instanciar una clase?(71)

Preguntas abiertas

- ◆ Cuéntame que es un objeto, una propiedad y un método
- ◆ Que son las librerías de clase (pág. 81)
- ◆ qué es la filosofía WORE? Pág. 84 (PAG 61 DEL LIBRO)
- ◆ Por qué el PHP se considera el numero uno en el desarrollo de páginas web
pág. 85
- ◆ Que es el esquema MVC (pág. 94)
- ◆ une cada lógica con su modelo
- ◆ Dime para qué sirve el NEW (pág. 52 del libro)
- ◆ Dime que es el THIS (pág. 75)

Preguntas abiertas

- ◆ que es la persistencia de objetos (pág. 76)
- ◆ qué es la herencia múltiple en la programación orientada a objetos (pág. 79)
- ◆ para que se utiliza la clase abstracta (pág. 80)

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Una ventajas de la centralización de software en aplicaciones web es que:

- ◆ facilitan las tareas de mantenimiento y actualización
- ◆ Evita tener copias de la aplicación en cada uno de los clientes, como puede suceder en aplicación de escritorio, herramienta ofimática....
- ◆ Al estar distribuido a través de la red, favorece las actualizaciones

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Pretendemos crear páginas web dinámicas , por lo que debemos utilizar mecanismos dinámicos para la creación de las mismas.

Estos mecanismos intentan que con un mismo código se puedan generar diferentes respuestas, según los parámetros seleccionados por el usuario.

Hay tres tipos de sitios web dinámicos:

- ◆ Los primeros, donde el dinamismo se encuentra en el lado del cliente
- ◆ Los segundos donde el dinamismo se encuentra en el nivel del servidor
- ◆ Los terceros divididos entre el cliente el servidor

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

En las tecnologías usadas del lado del servidor, se exige que devuelvan a los cliente respuestas en forma de código HTML. Esto evita que la máquina de el cliente, no tenga que tener un plug-in instalado.

Generalmente cualquier petición de un cliente a servidor, se trata de una página web, cuando además esta información incluye datos (ej. en un formulario), Estos datos pueden enviarse codificados en la propia URL. EN otras ocasiones , esa información traída del cliente, será utilizada para mantenimiento de sesiones de usuario (los cookies)

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- ◆ Esos datos del cliente se envían al servidor.
- ◆ Y el servidor podrá procesarlos y a su vez:
 - ◆ Acceder a bases de datos
 - ◆ Envío de mensajes a otras máquinas
 - ◆ Acceso a otros servidores o servicios web disponibles en la red

Después se debe generar dinámicamente una respuesta para enviársela al cliente, normalmente en formato HTML

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- ♦ Ventajas más destacables de las aplicaciones web desarrolladas de esta forma:
 - ♦ **Accesibilidad** desde cualquier punto de internet
 - ♦ **Fácil** mantenimiento : no será necesario distribuir el código de las aplicaciones ni sus actualizaciones
 - ♦ **Seguridad**: el código no puede manipularlo el usuario , al que solo le llega la respuesta
 - ♦ **Escalabilidad** Si utilizamos arquitecturas multicapa nos puede resultar relativamente sencillo ampliar un número de clientes a los que puede dar servicio la aplicación

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- ◆ Al igual que en el lado del cliente, en el lado del servidor se han dado dos tendencias:
 - ◆ La primera tiende a facilitar la creación de las página web a través de la utilización de lenguajes de script para servidor : en este caso el código está dentro de la página web que se ejecutara en el servidor y se diferencia lb código HTML por un grupo de tags que diferencia el código HTML del código script de servidor
 - ◆ Es segundo tipo de tecnologías tiende a crear pequeños elementos software los cuales son ejecutados y dan como resultado una página HTML

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- ◆ Vamos a profundizar en las tecnologías más populares dentro de esta área:
 - ◆ CGI COMMON GATEWAY INTERFACE
 - ◆ SERVLETS
 - ◆ LENGUAJES DE SCRIPT DEL LADO SERVIDORR

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- ◆ Vamos a profundizar en las tecnologías más populares dentro de esta área:
 - ◆ CGI COMMON GATEWAY INTERFACE
 - ◆ Tecnología que especifica un estándar para transferir datos entre el cliente y un programa residente en el servidor
 - ◆ Permiten al servidor web interactuar con aplicaciones para realizar
 - ◆ consultas a bases de datos
 - ◆ Búsqueda de documentos
 - ◆ Generadores de email automático
 - ◆ Comercio electrónico
 - ◆ Procesamiento de solicitudes entre otra acciones

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- ◆ El funcionamiento de un CGI se puede resumir en los siguientes pasos:
 1. El cliente acumula datos por ejemplo en un formulario HTML, y los prepara para su uso por la aplicación CGI enviándolos al servidor a través de la URL, usando el protocolo HTTP
 2. El servidor lee el URL que acompaña a la petición, determina cual es la aplicación CGI que debe ejecutar y la activa
 3. El servidor pasa la información de la URL a la aplicación CGI
 4. El programa CGI procesa los datos de acuerdo a su funcionalidad y prepara la respuesta. Generalmente la respuesta se formatea como un documento HTML , de acuerdo al protocolo HTTP
 5. El programa CGI pasa la respuesta al servidor quien la redirige hacia el cliente

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- Los programas que maneja el CGI pueden estar compilados en diferentes lenguajes de programación. El más popular para el desarrollo de contenidos web es el lenguaje Perl de distribución gratuita, aunque también es posible en Java, o Visual Basic
- Dado que lanzar un proceso nuevo cada vez que se envía un solicitud del usuario puede resultar costoso, varias compañías han desarrollado lo que se denominan extensiones de servidor
las extensiones de servidor son API que extienden la funcionalidad básica del servidor web para manejar tareas dinámicas.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

♦ SERVLETS

programa escrito en Java que se ejecuta en el servidor (los applets en navegador)

- ♦ Un servlet tiene preestablecido dos interfaces, una para la petición y otra para la respuesta, de esa manera los datos siempre son recibidos y devueltos de la misma forma , lo que facilita la creación de los mismos
- ♦ Cada servlet queda en la memoria del servidor web como una única instancia, la cual se carga la primer vez que llamamos al servlet

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ LENGUAJES DE SCRIPT DEL LADO SERVIDOR

- ◆ Fueron creados para crear páginas web dinámicas, mediante un código que será interpretado en el interior de las páginas HTML
- ◆ Se intercalan fragmentos de código dentro de los documentos HTML
- ◆ Los fragmentos de HTML proporcionan la parte estática , y los lenguajes de script la parte dinámica
- ◆ Suelen comenzar y finalizan el bloque por un scriptlet `<? O <%`
- ◆ Los lenguajes de script pueden ser clasificados en :
 - ◆ Lenguajes interpretados como, por ejemplo, PHP o ASP
 - ◆ Lenguajes pre-compilados como JSP o ASP.NET

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

LENGUAJES DE SCRIPT DEL LADO SERVIDOR

PHP(1994)

- Lenguaje de script interpretado en el lado del servidor , para crear páginas web dinámicas embebidas en documentos HTML y ejecutadas en el servidor
- De distribución libre
- Se ha convertido en un estándar para la creación de páginas web dinámicas . Hoy en día de más auge
- No necesita ser compilado para interpretarse
- Solo necesita que su servidor (Apache o IIS) tenga instaladas las librerías PHP
- Capacidad de conexión con la mayoría de servicios de bases de datos como MySQL, PostgreSQL, Oracle, SQL Server....
- Son archivos con extensión .php
- Bloque delimitado por <?php ?>
- Archivo php.ini contiene casi todos los detalles de configuración de PHP, además ese es el archivo desde donde se activan las diferentes librerías que podemos usar desde nuestras páginas PHP

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ LENGUAJES DE SCRIPT DEL LADO SERVIDOR

◆ ASP, ACTIVE SERVER PAGES

- ◆ Tecnología creada por Microsoft para desarrollar aplicaciones web dinámicas
- ◆ Tecnología ligada al servidor de aplicaciones Internet Information Server(IIS)de Microsoft
- ◆ Una pagina ASP No necesita ser compilada para ejecutarse
- ◆ los archivos tienen extensión .asp
- ◆ El lenguajes interpretado utilizados para crear páginas ASP es VBScript
- ◆ El código ASP se escribe entre lo símbolos <% y%>
- ◆ inconvenientes

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ LENGUAJES DE SCRIPT DEL LADO SERVIDOR

◆ ASP, ACTIVE SERVER PAGES

- ◆ Inconvenientes de página ASP que le hicieron estancarse en el 2002:
 - ◆ Código excesivamente extenso, poco legible y difícil de mantener al estar mezclado con HTML
 - ◆ Configurar una aplicación ASP grande puede ser una labor realmente compleja ,
 - ◆ presenta también problemas de eficiencia en cuanto al servidor web
 - ◆ Al estancarse, Microsoft lleva a ASP hacia la plataforma .Net y crea ASP .Net, lenguaje totalmente diferente.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

LENGUAJES DE SCRIPT DEL LADO SERVIDOR

JSP, Java Server Pages,

- Funcionamiento similar a ASP o PHP
- Permite generar documentos HTML o XML de forma dinámica combinando plantillas estáticas con el contenido dinámico
- EN JSP , se puede encargar a componente instalado en el servidor, la generación del contenido dinámico.
- Estos componentes se llaman JavaBeans, y requieren tener instalado un servidor de aplicaciones como Tomcat o cualquiera de los muchos servidores de aplicaciones existentes que se ajustan al estándar J2EE (Java 2 Enterprise Edition)
- Sintaxis de un scriptlet, es decir de bloque Java, dentro de HTML:
 - se utiliza <%conjunto de instrucciones %>
 - Las instrucciones simples deberán terminar en punto y coma

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ LENGUAJES DE SCRIPT DEL LADO SERVIDOR

◆ ASP. Net

- ◆ Teniendo en cuenta los inconvenientes de ASP, Microsoft insertó ASP en su plataforma .Net.
- ◆ La plataforma .Net pretende crear un marco de trabajo para el desarrollo rápido de aplicaciones
- ◆ Los archivos creados en ASP.Net tienen la extensión .aspx
- ◆ La diferencia con ASP:
 - ◆ si queremos crear páginas en ASP.Net podemos utilizar cualquiera de los lenguajes admitidos en la plataforma .Net
 - ◆ Uso de controles de servidor : objetos programables en el lado del servidor que suelen representar a un elemento de la interfaz gráfica como por ejemplo una caja de texto o una imagen
- ◆ Al igual que ASP la sintaxis de ASP.Net permite utilizar los delimitadores `<% %>`
- ◆ Cuando un cliente hace una petición de un archivo .aspx ASP.Net analiza y compila el archivo a una clase del entorno de trabajo .Net
- ◆ Se compila la primera vez que es accedido, y es la instancia compilada la que se reutiliza en las sucesivas peticiones

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ CONTROLES DE SERVIDOR



Los controles de servidor web ASP.NET son objetos de páginas web ASP.NET que se ejecutan cuando se solicita la página



Participan en la ejecución de la página , y producen sus propias etiquetas.



La principal ventaja de los controles de servidor es que permiten a los desarrolladores obtener comportamientos complejos a partir de componentes sencillos , reduciendo así la cantidad de código necesaria para la creación de un pagina web dinámica



Muchos controles de servidor Web son similares a los conocidos elementos HTML, como botones y cuadros de texto. Otros controles abarcan comportamiento complejo, como los controles de un calendario, y controles que puede usar para conectar a orígenes de datos y datos de visualización.



ASP.NET también proporciona controles de servidor habilitados para AJAX. Estos controles están compuestos por código de servidor y de cliente que se integra para generar un comportamiento de cliente enriquecido. Cuando se agrega un control AJAX a una página web ASP.NET, la página envía automáticamente el script de cliente de soporte al explorador para la funcionalidad de AJAX. Puede proporcionar también código de cliente adicional para personalizar la funcionalidad de un control, pero no es necesario.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- ◆ Cuando crea páginas Web ASP.NET, puede utilizar estos tipos de controles:
 - ◆ Controles de servidor HTML Elementos HTML expuestos al servidor para que se puedan programar.
 - ◆ Controles de servidor Web controles de formulario como botones y cuadros de texto, controles con fines especiales como un calendario, menús y un control de vista de árbol.
 - ◆ Controles de validación: permiten comprobar un campo necesario, su adecuación a un valor o un modelo de caracteres concreto, comprobar que un valor se encuentra en un intervalo predefinido, etc. P
 - ◆ Controles de usuario: barras de herramientas y otros elementos reutilizables.

Nota: También se utilizan controles diseñados específicamente para dispositivos móviles.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.2 CARÁCTERÍSTICAS GENERALES

- La programación en el entorno del servidor en el caso de los lenguajes embebidos, se centra en la inserción de código propio en páginas HTML
- Cada lenguaje utiliza una serie de etiquetas para delimitar los fragmentos de código que han de ser procesados por el servidor web.
- La información embebida en esas etiquetas contiene código fuente escrita de acuerdo a las características del lenguaje escogido.

A continuación haremos un repaso de características generales de los lenguajes de programación de aplicaciones web en el lado del servidor abarcando: definición de variable y los tipos de datos asignados, los operadores y estructuras de control y aspectos relacionados con el soporte, la orientación a objetos de estos lenguajes...

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.3 VARIABLES Y TIPOS DE DATOS.

- En cualquier lenguaje de programación web, los datos manejados se almacenan en variables.
- Se trata de almacenes temporales de datos que permiten gestionar los datos utilizados por la aplicación web durante el flujo de ejecución de una página determinada
- Toda variable tendrá asociado un tipo de datos

1. DEFINICION Y USO

- En PHP se identifican por el símbolo \$ seguido del nombre de la variable
- En PHP las variables no necesitan ser declaradas explícitamente y además no tienen un tipo definido hasta que no se les asigna un valor
- Y este valor, y el tipo, puede cambiar a lo largo de su vida

2. TIPOS DE DATOS Y VARIABLES

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.3 VARIABLES Y TIPOS DE DATOS.

- En cualquier lenguaje de programación web, los datos manejado se almacenan en variables.
- Se trata de almacenes temporales de datos que permiten gestionar los datos utilizados por la aplicación web durante el flujo de ejecución de una página determinada
- Toda variable tendrá asociado un tipo de datos

1. DEFINICION Y USO

- En PHP se identifican por el símbolo \$ seguido del nombre de la variable
- En PHP las variable no necesitan ser declaradas explícitamente y además no tienen un tipo definido hasta que no se les asigna un valor
- Y este valor, y el tipo, puede cambiar a lo largo de su vida
- Una de las mayores fortalezas de PHP es que al cambiar de valor cambia su tipo sin necesidad de definirlo. Esto hace que sea más flexible, aunque hay que tener precaución porque a veces conlleva riesgos.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.3 VARIABLES Y TIPOS DE DATOS.

- Reglas que debe cumplir el nombre de una variable
 - Debe empezar por una letra o guión bajo
 - Solo puede contener caracteres alfanuméricos y guiones bajos
 - No debe contener espacio en blanco (sustituye por guion bajo o por primera letra de la segunda palabra en mayúscula)
- IMPORANTE: Es case-sensitive : distingue entre mayúscula y minúscula

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.3 VARIABLES Y TIPOS DE DATOS.

1. TIPOS DE DATOS Y VARIABLES

Tipos de datos que podemos diferenciar en PHP:

1. Escalares pueden contener un único valor cada vez
 1. Boolean las mas sencillas. Solo pueden contener los valores true y false
 2. Integer Números enteros
 3. Float números reales con una parte entera y otra decimal
 4. String texto (entre comilla doble o simple)
2. Compuestos formadas por múltiples valores escalares
 1. Array
 2. Object

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.3 VARIABLES Y TIPOS DE DATOS.

1. TIPOS DE DATOS Y VARIABLES

Tipos de datos que podemos diferenciar en PHP:

1. Especiales
 1. Null
 2. Resource
2. Seudo-tipos
 1. Mixed
 2. Number
 3. callback

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.3 VARIABLES Y TIPOS DE DATOS.

1. TIPOS DE DATOS Y VARIABLES

Constantes:

aquel contenido que no varia durante la ejecución del código

Se definen en PHP mediante la función `define()` que admite dos argumentos obligatorios, un nombre y un valor que se asociarán a partir de ese momento al nombre

Ojo!! Las constantes no llevarán la variable \$ delante del nombre

Su valor no puede cambiarse una vez ha sido definida

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

💧 4.4 SENTENCIAS CONDICIONALES

Estructuras de control que permiten decidir el flujo de ejecución de un programa, es decir el orden el que las instrucciones de un programa se van a ejecutar

El código deja de ejecutar las instrucciones de manera secuencial , una detrás de otra para poder definir caminos alternativos dependiendo de si se cumplen las condiciones establecidas por el programador

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.4 SENTENCIAS CONDICIONALES

Sentencia if

define dos flujos de ejecución dependiendo si se cumple o no la condición establecida por el programador. Si se cumple la condición se ejecutara una instrucción o un grupo de instrucciones. Si no se cumple la condición se ejecutarán otras instrucciones distintas

Ojo!!

1. el uso de las llaves es opcional cuando un bloque de sentencias contiene únicamente una instrucción
2. Para separar una instrucción de otra se utiliza el símbolo ;
3. Podemos definir varios if anidados que nos permitan evaluar varias condiciones previas antes de ejecutar las instrucciones correspondientes

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.4 SENTENCIAS CONDICIONALES

Smith o select case

Se usan cuando dependiendo del valor que toma una variable o expresión , necesitamos que se ejecute un conjunto de instrucciones distintas para cada uno de los valores que pueda tomar ((puede sustituir al if. Altamente recomendable)

1. **Funcionamiento:** Se evalúa el valor de la expresión y se compara dicho valor con cada uno de los casos
2. Cuando se encuentra con el caso con el que coincide, se ejecutan las instrucciones contenidos dentro del caso correspondiente
3. Si al evaluar todos los casos no coincide con ninguno, se ejecutan las instrucciones definidas en el bloque por defecto

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.5 BUCLES

- Ejecuta de forma reiterativa una instrucción o grupo de instrucciones.
- Estas instrucciones se pueden ejecutar un número determinado o indeterminado de veces dependiendo del bucle que utilicemos para ello
 - **WHILE.** Permite ejecutar un numero indeterminado de veces una instrucción o grupo de instrucciones, mientras se cumpla la condición
 - **DO-WHILE O DO..LOOP WHILE** igual que el anterior, pero con la salvedad de que siempre se ejecuta al menos una vez , se cumpla o no la condición
 - **BUCLE DO UNTIL....LOOP** (propio del lenguaje asp). Similar al bucle While. La condición se evalúa al principio de cada iteración
 - **BUCLE DO LOOP UNTIL.** Equivalente al Do Whiles en ASP
 - **BUCLE FOR O FOR...NEXT.** Se ejecuta la **instrucción** un numero determinado d veces

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.5 BUCLES

- ◆ **BUCLER FOREACH** para iterar de forma sencilla por los elementos de un array .
Este bucle solo funciona con arrays o matrices
- ◆ **SENENCIA BREACK**
para forzar la interrupción de la ejecución de un bucle como por ejemplo ocurre cuando utilizamos los switches
- ◆ **SENTENCIA CONTINUE** Se utiliza para saltar el resto de la iteración de un bucle y continuar con la evaluación de la condición y la siguiente iteración.
Sintaxis: `continúe[numero]`

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.6 GESTION DE ERRORES

- Se trata de funciones que se ocupan del manejo y registro de errores. Le permiten definir sus propias reglas de manejo de errores, así como modificar la manera en que los errores pueden ser registrados.
- Las funciones de notificación de errores le permiten personalizar qué niveles y tipos de errores se dan, abarcando desde simples avisos hasta funciones personalizadas devueltas al originarse un error.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.6 GESTION DE ERRORES

Errores y Registro

Valor	Constante	Descripción	Nota
1	E_ERROR (integer)	Errores Fatales en tiempo de ejecución. Éstos indican errores que no se pueden recuperar, tales como un problema de asignación de memoria. La ejecución del script se interrumpe.	
2	E_WARNING (integer)	Advertencias en tiempo de ejecución (errores no fatales). La ejecución del script no se interrumpe.	
4	E_PARSE (integer)	Errores de análisis en tiempo de compilación. Los errores de análisis deberían ser generados únicamente por el analizador.	
8	E_NOTICE (integer)	Avisos en tiempo de ejecución. Indican que el script encontró algo que podría señalar un error, pero que también podría ocurrir en el curso normal al ejecutar un script.	
16	E_CORE_ERROR (integer)	Errores fatales que ocurren durante el arranque inicial de PHP. Son como un E_ERROR , excepto que son generados por el núcleo de PHP.	
32	E_CORE_WARNING (integer)	Advertencias (errores no fatales) que ocurren durante el arranque inicial de PHP. Son como un E_WARNING , excepto que son generados por el núcleo de PHP.	

funciones personalizadas devueltas al originarse un error.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.6 GESTION DE ERRORES

64	E_COMPILE_ERROR (<u>integer</u>)	Errores fatales en tiempo de compilación. Son como un E_ERROR , excepto que son generados por Motor de Script Zend.
128	E_COMPILE_WARNING (<u>integer</u>)	Advertencias en tiempo de compilación (errores no fatales). Son como un E_WARNING , excepto que son generados por Motor de Script Zend.
256	E_USER_ERROR (<u>integer</u>)	Mensaje de error generado por el usuario. Es como un E_ERROR , excepto que es generado por código de PHP mediante el uso de la función de PHP <u>trigger_error()</u> .
512	E_USER_WARNING (<u>integer</u>)	Mensaje de advertencia generado por el usuario. Es como un E_WARNING , excepto que es generado por código de PHP mediante el uso de la función de PHP <u>trigger_error()</u> .
1024	E_USER_NOTICE (<u>integer</u>)	Mensaje de aviso generado por el usuario. Es como un E_NOTICE , excepto que es generado por código de PHP mediante el uso de la función de PHP <u>trigger_error()</u> .

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.6 GESTION DE ERRORES

[trigger_error\(\)](#).

2048	E_STRICT (<u>integer</u>)	Habilítelo para que PHP sugiera cambios en su código, lo que asegurará la mejor interoperabilidad y compatibilidad con versiones posteriores de PHP de su código.	Desde PHP 5 pero no incluido en E_ALL hasta PHP 5.4.0
4096	E_RECOVERABLE_ERROR (<u>integer</u>)	Error fatal capturable. Indica que ocurrió un error probablemente peligroso, pero no dejó al Motor en un estado inestable. Si no se captura el error mediante un gestor definido por el usuario (vea también <u>set_error_handler()</u>), la aplicación se abortará como si fuera un E_ERROR .	Desde PHP 5.2.0
8192	E_DEPRECATED (<u>integer</u>)	Avisos en tiempo de ejecución. Habilítelo para recibir avisos sobre código que no funcionará en futuras versiones.	Desde PHP 5.3.0
16384	E_USER_DEPRECATED (<u>integer</u>)	Mensajes de advertencia generados por el usuario. Son como un E_DEPRECATED , excepto que es generado por código de PHP mediante el uso de la función de PHP <u>trigger_error()</u> .	Desde PHP 5.3.0
32767	E_ALL (<u>integer</u>)	Todos los errores y advertencias soportados, excepto del nivel E_STRICT antes de PHP 5.4.0.	32767 en PHP 5.4.x, 30719 en PHP 5.3.x, 6143 en PHP 5.2.x, 2047 anteriormente

funciones personalizadas devueltas al originarse un error.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.7 GESTION DE BIBLIOTECAS

- Algunas funciones son muy generales, y se repiten en cada construcción web, Por eso casi todos los lenguajes de programación han desarrollado su propio framework con el fin de facilitar la operativa en la creación de las páginas web
- Vamos a profundizar en conceptos como librería , interfaz de programación de aplicaciones o framework.
- Las librerías contienen un código y datos que pueden ser llamado desde otro programa principal. De esta form a se evita que cada programa principal desarrolle el código de la librería
- La interfaz de Programación de Aplicaciones es un conjunto de funciones y procedimientos que proveen de bibliotecas (librerías) para se utilizado por otro programa . La diferencia con la librería es que las funciones de una API acceden habitualmente a programas diferentes permitiendo así generar transparencia al programador

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.7 .1as tecnologías y librerías relacionadas con ASP

- Cada lenguaje de programación en el servidor ha desarrollado sus propias tecnologías , para maximizar las opciones y adaptarse a las necesidades del mercado web.
- Según W3C, el lenguaje ASP es capaz de
 - Dinamizar, editar, cambiar añadir cualquier contenido de una página web
 - Responder a las consultas d e los usuario a lo sitios de los formularios HTML que se han enviado
 - Acceder a datos y a bases de datos, devolviendo resultados al navegador
 - Personalizar las aplicaciones web para que sean más útiles ara usuarios individuales
 - Es una tecnología veloz y sencilla. Además garantiza la seguridad, ya que el código no se ve en el navegador. Si se programa a de una forma correcta minimiza el tráfico de Red

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.7 .las tecnologías y librerías relacionadas con ASP

ASP dispone de una tecnología que implementa alguno de los elementos HTML en el servidor . Con esta tecnología los elementos HTML de servidor, en vez de ser sujetos pasivos esperando a que una condición ASP los deje aparecer , tiene n una propiedades intrínsecas que los otorga autonomía. A estos elementos se les llama controles.

- Los controles de servidor en ASP son objetos de las páginas web que se ejecutar cuando el usuario solicite una página. Muchos de estos son similares a los elementos habituales en HTML como botones, cajas de texto, etc. .
- Además de encontrarnos los elementos HTML, los controles ASP nos proporcionan otros más complejos, como calendarios o conexiones a bases de datos
- El framwork que Microsoft ahí facilitado para hacer uso de esos controles es ASP. NET. Por tanto estos controles permiten manejar efectos en el servidor dependiendo de l iteración del usuario

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.7 .las tecnologías y librerías relacionadas con ASP

Cómo estos eventos son manejados desde el servidor, es necesario acceder a las propiedades, métodos o eventos de los elementos en el código servidor.

Existen varios controles de servidor en ASP.Net:

- HTML, SERVER CONTROLS. Estos controles son los mismos que los de HTML. La diferencia es que estos son a nivel servidor
- ASP.NET Web Forms Control. También como los de HTML en el lado del servidor. Ej. buttons, textbox....
- ASP Net List Controls. Control que permite mostrar el contenido de las bases de datos
- ASP.Net Mobile Controls. igual que los dos anteriores, pero la diferencia reside en que se aplican a los dispositivos cliente, como por ejemplo un dispositivo móvil
- ASP.Net Validation Controls Controles de validación. Permiten validar en el servidor entradas a través de la iteración que el usuario tiene con la aplicación. La ventaja que presenta es la variedad de casos para validar.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.7 .las tecnologías y librerías relacionadas con ASP

ASP dispone de una tecnología que implementa alguno de los elementos HTML en el servidor . Con esta tecnología los elementos HTML de servidor, en vez de ser sujetos pasivos esperando a que una condición ASP los deje aparecer , tiene n una propiedades intrínsecas que los otorga autonomía. A estos elementos se les llama controles.

- 🔹 [PowerCollections](#) - Es una librería de colecciones genéricas .

- 🔹 [StructureMap](#) – de m fácil uso

- 🔹 [Log4Net](#) - Desde que conocí Log4net, después de una -muy mala, malísima vamos- experiencia con otro tipo de librerías, también Open Source, me parece la apuesta más fácil y eficaz para el logging de aplicaciones.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

TECNOLOGÍAS Y LIBRERIAS PHP

- la tecnología PHP es un de los lenguajes de generación de paginas web más populares , El estándar php indica que esto es por:
- independencia de su plataforma : funciona en diferentes plataformas (Windows, Linux...
- libertad en tipo de licencia
- rapidez de aprendizaje
- Es compatible con casi todos los servidores utilizan en la actualidad (apache, IIS...
- se puede descargar gratis
- cualquier cosa que se te ocurra hacer, ya la ha hecho alguien antes accesible a través de librerías.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.7 .Tecnologías y librerías relacionadas con JSP

AL hablar de JSP, nos referimos a páginas Java que se ejecutan en el servidor.

Es una tecnología que nos permite ejecutar código Java en el servidor a través de scripts, para generar páginas web dinámicas.

En Java existen dos tipos de fichero principales que se ejecutan en el servidor. Ambos tipos de fichero pueden realizar las mismas operaciones

- Uno de ellos es JSP el cual facilita la presentación de los documentos
- El otro es el servlet que provee de una mayor flexibilidad en la programación de la lógica de negocio.

Ventaja de tecnologías Java relacionadas con el servidor: Que se pueden ejecutar en múltiples plataformas.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.7 .Tecnologías y librerías relacionadas con JSP

Diferencias entre JSP y servlets

- Objetivo JSP (java Server Page): Estructurar la interfaz de usuario de una forma clara y ordenada
- Los Servlets son programas que se ejecutan en el lado del servidor para generar páginas web. Estos actúan normalmente a partir de parámetros enviados a una petición desde el navegador . Por su forma facilitan en su implementación la programación del modelo de negocio

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

Hablemos de el **control de usuarios**

- Las aplicaciones web están diseñadas para que usuarios con distintos niveles de acceso puedan realizar diversas operaciones, pero ¿cómo podemos organizar dichos niveles de acceso? A través de **perfiles o roles**.
- Esto implica que un usuario cualquiera pueda entra y utilizar nuestra aplicación web, según un rol.
- El rol es una de las características del usuario y es la propiedad que establece cuales son las tareas o procesos que el usuario puede ejecutar.
Se trata de crear un control de acceso jerárquico en el cual un usuario tiene un rol, un rol contiene varias tareas y una tarea es ejecutada a través de varios procesos.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

ACL: LISTA DE CONTROL DE ACCESO

- son listas de condiciones que permiten designar permisos de acceso a cualquier elemento del sistema o directorio en general.
- En función de estas condiciones, se concede o deniega el acceso a la modificación de las propiedades de estos elementos a los diferentes usuarios o procesos del sistema. Es, por tanto, un mecanismo de seguridad del sistema.
- En el caso de referirnos, por ejemplo, al sistema de archivos, las ACL proporcionan un nivel adicional de seguridad a los archivos, ya que extiende el clásico esquema de permisos. En el caso de referirnos, por ejemplo, a routers las ACL establecen condiciones que se aplican al tráfico que viaja a través de la interfaz del router. Es decir, las ACL informan al router de qué tipo de paquetes debe aceptar y cuáles rechazar

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

Las ACLs permiten asignar permisos a usuarios o grupos concretos. Por ejemplo, se pueden dar ciertos permisos a dos usuarios sobre unos archivos sin necesidad de incluirlos en el mismo grupo.

Los archivos y directorios tienen conjuntos de permisos configurados para el propietario del archivo, el grupo al que pertenece el dueño del archivo y los otros usuarios del sistema. Sin embargo, estos permisos tienen sus limitaciones. Por ejemplo, no se pueden configurar diferentes permisos para usuarios diferentes. Para solucionar este problema se crearon las Listas de Control de Acceso (Access Control Lists, ACLs).

El concepto de ACL permite un control más preciso que los permisos del archivo por sí solos, al dar al propietario de un objeto la capacidad de conceder o denegar accesos usuario por usuario.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

Este mecanismo está disponible en la mayoría de Unix (Solaris, AIX, HP-UX, etc.), mientras que en otros que no lo proporcionan por defecto, como GNU/Linux, pero puede instalarse como un paquete adicional (acl).

GNU/Linux soporta 2 tipos básicos de ACL:

ACL estándar: ACL de control de directorios y ficheros. Contiene únicamente las entradas para los tipos propietario, grupo propietario y otros, que corresponden a los bits de permiso convencionales para archivos y directorios.

ACL extendida: ACL que incluye más elementos. Una entrada máscara y puede contener al mismo tiempo distintas entradas de los tipos usuario nombrado y grupo nombrado.

Las ACLs se pueden configurar:

- Por usuario

- Por grupo

- A través de la máscara de permisos

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

AUTENTICACION DE USUSARIOS : OPENID Y OAUTH

Otra tendencia que podemos observar actualmente en el desarrollo de aplicaciones para Internet es la denominada autenticación Universal. Con el aumento de servicios on line (Hotmail, Gmail,, Facebook...) ha aumentado la colección de nombres de usuario y contraseña. Normalmente cada vez que un usuario desea probar un nuevo servicio en la Web, el primer paso es registrarse en el sitio Web de dicho servicio. Mucha veces el nombre de usuario que se utiliza usualmente se ven forzados a inventar nuevos nombres de usuario q por desgracia se suelen olvidar.

Para solucionar este problema han surgido iniciativas como OpenId (iniciativa de identificación digital, que permite al usuario entrar a una página Web, pudiendo ser verificado por cualquier otro servidor que soporte este protocolo) y OAuth(protocolo abierto para disponer de una API segura de autorización a través de un método estándar y simple para aplicaciones Web y de escritorio.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

El funcionamiento percibido por el usuario es simple en ambos casos. El servicio a consumir en el que el usuario no se ha registrado se comunica con el servicio validador, es decir el servicio en el cual el usuario ya está registrado y al que se le positiva que realice la comprobación del usuario

Entre las ventajas de este sistema podemos enumerar las siguientes:

Con una única cuenta podemos manejar múltiples servicios

Limitamos la cantidad de información que nos vemos obligados a promocionar según a qué proveedor de servicios incrementando nuestra privacidad.

Estas herramientas de autenticación de usuarios cada vez están más extendidas, formando parte ya de muchos frameworks y CMS como Joomla, Drupal, WordPress.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

AUTENTICACION DE USUSARIOS : OPENID Y OAUTH

EJEMPLO: UNA GUIA DE IMPLEMENTACION DE OAUTH

- En este proceso intervienen tres actores:
- El servicio al que el usuario quiere acceder usando una cuenta externa al cual se le llama consumidor,
- el servicio de autenticación externo al que accederá a través de OAuth al que se le llama proveedor de identidad y
- el usuario.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

De manera general el procedimiento a implementar es el siguiente:
El consumidor debe registrarse como tal en el proveedor del servicio se autenticación. Por ejemplo en el caso de querer usar el servicio de autenticación de Twiter, el primer paso es conectarse con la cuenta de Twiter que queremos que sea la propietaria de la aplicación, es decir entramos con la cuenta de Twiter de consumidor y registramos una nueva aplicación (en Twiter podemos encontrar esta opción bajo el menú configuración > conexiones en el apartado para desarrolladores con un link (registre una nueva aplicación) donde luego deberemos rellenar un formulario. El proveedor de servicios nos proporciona los datos necesarios para crear nuestra firma digital y las URL donde se encontrarán los servicios

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

Cualquier comunicación mediante el protocolo OAuth se iniciara con una llamada denominada request token o petición de token. En este paso el consumidor y el proveedor del servicio se autentican mutuamente mediante una firma y acuerdan un token que utilizaremos para autenticar al usuario . Un token es un código único de carácter temporal que sirve para identificar normalmente una sesión . El token o está asociado a ningún usuario hasta que el usuario lo valida una vez lanzado el request token el proveedor de identidad devuelve un `oauth_token` y un `auto token secret` al consumidor . Estos dos parámetros deben ser guardados para utilizarlos más adelante. Preferiblemente en la sesión de usuario o en una cookie.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

El siguiente paso es redirigir al usuario a la página de autenticación del profesor de identidad, pasando el token como parámetro una vez en la página del proveedor de servicio , el usuario se autentica validando el `oauth_token`. Tras validarlo el `oauth_token` enviado por el consumidor y los datos que el usuario ha ingresado , el proveedor del servicio redirige nuevamente al usuario a la página base que se ha especificado en el momento de dar de alta la aplicación del consumidor en el proveedor del servicio o a la especificada en el parámetro opcional `OAuth-callback` el consumidor recoge al usuario en la URL dada en el callback junto con el token de confirmación de identidad. el token con que el proveedor del servicio envía de vuelta al usuario a la aplicación del consumidor es el mismo que ya se tenía almacenado en la sesión de usuario. Comparando ambos tokens se verifica si el usuario es quien dice ser y si es así, se le da acceso a la aplicación

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

. SESIONES WEB E IDENTIFICADORES DE SESIÓN Las páginas web tienen «memoria», reconocen las acciones que el usuario ha realizado anteriormente como, por ejemplo, si se ha registrado, qué elementos ha visitado o las compras añadidas a la cesta. Dicho de otro modo, establecen una sesión con el internauta. Si HTTP, el protocolo con el que se interactúa con las páginas web, no está orientado a conexión, ya que por sí mismo no proporciona manera de almacenar las acciones que el navegante realiza en una página web¹, ¿cómo es posible mantener una sesión web? Para mantener las sesiones web el navegador y el servidor web comparten un identificador único que el navegador web incluye en cada petición HTTP o HTTPS realizada al portal (generalmente mediante cookies). De este modo, por medio de este identificador, el servidor web puede reconocer que la petición que recibe pertenece a una determinada sesión, almacenar la información de esta petición que le interese y responder a ella según la información almacenada anteriormente. Normalmente, al autenticarse un usuario en una página web o portal, se incluye en la información de sesión el identificador del usuario; de este modo, en peticiones HTTP posteriores el portal reconoce, a través del identificador de sesión, a qué usuario corresponde esa petición y puede asociarle las acciones realizadas y personalizar su contenido. Del mismo modo, al salir un usuario de un portal o aplicación web, éste cierra la sesión web. En la siguiente imagen se puede apreciar un «diálogo» entre el navegador y el servidor web capturado con el complemento de Firefox Live HTTP headers, también se puede utilizar ieHTTPHeaders para Internet Explorer:

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.8 GESTION DE LA CONFIGURACION Y LA SEGURIDAD

. SESIONES WEB E IDENTIFICADORES DE SESIÓN Las páginas web tienen «memoria», reconocen las acciones que el usuario ha realizado anteriormente como, por ejemplo, si se ha registrado, qué elementos ha visitado o las compras añadidas a la cesta. Dicho de otro modo, establecen una sesión con el internauta. Si HTTP, el protocolo con el que se interactúa con las páginas web, no está orientado a conexión, ya que por sí mismo no proporciona manera de almacenar las acciones que el navegante realiza en una página web¹, ¿cómo es posible mantener una sesión web? Para mantener las sesiones web el navegador y el servidor web comparten un identificador único que el navegador web incluye en cada petición HTTP o HTTPS realizada al portal (generalmente mediante cookies). De este modo, por medio de este identificador, el servidor web puede reconocer que la petición que recibe pertenece a una determinada sesión, almacenar la información de esta petición que le interese y responder a ella según la información almacenada anteriormente. Normalmente, al autenticarse un usuario en una página web o portal, se incluye en la información de sesión el identificador del usuario; de este modo, en peticiones HTTP posteriores el portal reconoce, a través del identificador de sesión, a qué usuario corresponde esa petición y puede asociarle las acciones realizadas y personalizar su contenido. Del mismo modo, al salir un usuario de un portal o aplicación web, éste cierra la sesión web. En la siguiente imagen se puede apreciar un «diálogo» entre el navegador y el servidor web capturado con el complemento de Firefox Live HTTP headers, también se puede utilizar ieHTTPHeaders para Internet Explorer:

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.9 TRANSACCIONES Y PERSISTENCIA

- Hasta aquí hemos visto aspectos referentes a las pruebas a las que hay que someter a las aplicaciones web durante su desarrollo, la autenticación de lo usuario y el control de sesiones.
- Ahora vamos a ver:
 - cómo gestionar la comunicación de estas aplicaciones web con una base d e datos
 - cómo utilizar la información recuperada de la base de datos,
 - cómo realizar transacciones
 - y cómo accederá a la base de datos de manera “serializable” para que la base de datos no se quede en ningún momento inconsistente (por el acceso e varias personas a la vez
 - Como obtener información de otro tipo de fuentes que no sean la propia base de datos

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.9 TRANSACCIONES Y PERSISTENCIA

ESTABLECIMIENTO DE CONEXIONES

- A la hora de interactuar con una base de datos , lo primero que tenemos que hacer es establecer la conexión entre nuestra aplicación web y esta para que posteriormente podamos ejecutar las sentencias SQL
- Diferencias entre los distintos lenguajes para conectarse a una base de datos (mirar el libro)

Php

- se utiliza la función mysqli utilizada a partir de la version 4.1 y la mysql que se utilizan en versiones anteriores.
- En ambas tenemos que facilitar el servidor, el usuario y la contraseña
La diferencia entre ambas es que mysqli, permite añadir al principio la base de datos, o hacerlo más tarde el nombre de la base de datos al final. Con la versión anterior primero hay que conectarse al servidor y después seleccionar la base de datos

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.9 TRANSACCIONES Y PERSISTENCIA

ESTABLECIMIENTO DE CONEXIONES

JSP

1. Requiere importar el paquete `java.sql` que contiene todos los métodos y clases relacionados con el manejo de las bases de datos
2. Se crea una instancia de los drivers específicos de la base de datos con la que vamos a trabajar .
3. A continuación utilizamos esos drivers para conectarnos con la base de datos mediante el método `DriversManager.getConnection()`. Este método recibe cómo parámetro una URL , y opcionalmente podemos especificar el nombre de usuario y la contraseña de la base de datos a la que queremos conectar

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

4.9 TRANSACCIONES Y PERSISTENCIA

ESTABLECIMIENTO DE CONEXIONES

ASP

1. Creamos una variable sobre la que llamaremos al método `server.CreateObject()`.
Este método crea una instancia de un objeto del servidor.
2. Después debemos crear la conexión especificando los drivers que se van a utilizar para ello.
3. También podemos especificar el puerto, el nombre de la base de datos , el usuario y la contraseña. Por último, también podremos especificar la ruta del servidor que contiene la base de datos.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ ACCESO A DIRECCIONES Y OTRAS FUENTES DE DATOS

Un directorio es una guía en la que se lista un conjunto de objetos así como información diversa acerca de cada uno de dichos objetos.

Los directorio permiten localizar información y, para ello, definen qué información se almacenará así como la forma en la que esta se organiza.

A nivel informático, un servicio de directorio es el conjunto de componentes hardware y /o software que trabajan de forma cooperativa para prestar un servicio de búsqueda e información acerca de recursos electrónicos como usuarios , impresoras, archivos....

LDAP (protocolo Ligero de Acceso al Servicio de Directorios. Es un protocolo a nivel de la aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar información diversa en un entorno red.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

DAP es un protocolo a nivel de aplicación, por lo tanto el cliente como el servidor deben implementar completamente la torre de protocolos OSI.

LDAP surge como una alternativa a DAP y utiliza TCP /IP en lugar de protocolos OSI con lo cual requiere menos recursos de red .

Además el modelo funcional de LDAP es más simple y ha eliminado opciones raramente utilizadas en X.500 Lite.

También LDAP organiza la información en un modo jerárquico usando directorios. Estos directorios puedan almacenar una gran variedad de información. Sin embargo en la mayoría de los casos LDAP se usa simplemente como un directorio telefónico virtual permitiendo a los usuarios acceder fácilmente a la información de contacto de otros usuarios

El LDAP va mucho más lejos que un directorio telefónico tradicional, Ya que es capaz de propagar su consulta a otros servidores LDAP por todo el mundo, proporcionando un repositorio global de información ad hoc.

Sin embargo en este momento LDAP se usa más dentro de organizaciones individuales publicas o privadas

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

LDAP Es un sistema cliente /servidor . el servidor puede ser implementado usando una gran variedad de bases de datos para guardar el directorio , cada uno optimizado para operaciones de lectura rápidas y de gran volumen.

Cuando una aplicación cliente se conecta a un servidor LDAP puede realizar funciones de consulta de un directorio o de modificación del mismo.

Si se realiza una consulta, el servidor puede contestar localmente o dirigir la consulta a otro servidor LDAP, que contenga la respuesta. Si la aplicación cliente está intentando modificar información en el directorio LDAP, el servidor verifica primero que el usuario tiene permiso para efectuar el cambio y después añade o actualiza la información.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

LA SEGURIDAD DE LA INFORMACION ALMACENADA EN EL DIRECTORIO ES UNO DE LOS ASPECTOS MÁS Importantes a la hora de trabajar con LDAP. Algunos directorios deben permitir el acceso público pero esto no implica que cualquier usuario pueda realizar todo tipo de operaciones.

De manera general se entiende que todos los usuarios pueden buscar información sobre otros usuarios, pero solo el administrador o el usuario involucrado debe tener permiso para modificar la información.

Es decir el directorio creado debe permitir la implementación de una política de seguridad que defina los permisos de cada usuario. De acuerdo a la herramienta utilizada, el directorio puede o no incorporar estas capacidades. En caso de no incorporarlas debe poder integrarse con un servicio de red fiable que proporcione estos servicios de seguridad.

Un método para gestionar estos roles puede ser a través de una ACL (Lista de Control de Acceso)

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ ESTRUCTURA DE DIRECTORIO LDAP

un directorio es un conjunto de objetos con atributos organizados de una manera lógica y jerárquica . Además de definir el protocolo de acceso al directorio , el estándar LDAP define cuatro modelos que permiten entender mejor el servicio de directorio.

- ◆ Modelo de información: describe la estructura de la información almacenada en el directorio LDAP.
- ◆ Modelo de nombres o de nombrado: describe como se organiza e identifica la información en el directorio LDAP
- ◆ Modelo funcional : describe qué operaciones pueden ser realizadas sobre la información almacenada en el directorio LDAP
- ◆ Modelo de seguridad: describe cómo puede protegerse la información contenida en el directorio LDAP frente a accesos no autorizados

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ ESTRUCTURA DE DIRECTORIO LDAP

Un árbol de directorio LDAP a veces refleja varios límites políticos, geográficos u organizacionales, dependiendo del modelo elegido. Los despliegues actuales de LDAP tienden a usar nombres de Sistema de Nombres de Dominio ([DNS](#) por sus siglas en inglés) para estructurar los niveles más altos de la jerarquía. Conforme se desciende en el directorio pueden aparecer entradas que representan personas, unidades organizacionales, impresoras, documentos, grupos de personas o cualquier cosa que representa una entrada dada en el árbol (o múltiples entradas).

◆ Habitualmente, almacena la información de [autenticación](#) (usuario y [contraseña](#)) y es utilizado para autenticarse aunque es posible almacenar otra información (datos de contacto del usuario, ubicación de diversos recursos de la red, permisos, [certificados](#), etc.). A manera de síntesis, LDAP es un protocolo de acceso unificado a un conjunto de información sobre una red

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

ESTRUCTURA DE DIRECTORIO LDAP

Visión general del protocolo

Un cliente inicia una sesión de LDAP conectándose a un servidor LDAP, preestablecido en el puerto TCP 389. El cliente luego envía una petición de operación al servidor, y el servidor envía respuestas. Con algunas excepciones, el cliente no necesita esperar una respuesta antes de enviar la siguiente petición, y el servidor puede responder en cualquier orden.:

Start TLS — usar la extensión Transport Layer Security (TLS) LDAPv3 para una conexión segura

Bind — autenticarse y especificar una versión del protocolo LDAP

Search — buscar y obtener entradas de directorio

Compare — probar si una entrada nombrada contiene un valor de atributo dado

Add — Añadir una nueva entrada

Delete — Borrar una entrada

Modify — Modificar una entrada

Modify Distinguished Name (DN) — Modificar o renombrar una entrada

Abandon — abortar una petición previa

Extended Operation — operación genérica usada para definir otras operaciones

Unbind — cerrar la conexión (no es el inverso de Bind)

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

ESTRUCTURA DE DIRECTORIO LDAP

El protocolo accede a directorios LDAP, que siguen la edición de 1993 del modelo X.500:

- Un directorio es un árbol de entradas de directorio.

- Una entrada consta de un conjunto de atributos.

- Un atributo tiene un nombre (un *tipo de atributo* o *descripción de atributo*) y uno o más valores. Los atributos son definidos en un *esquema* (véase luego).

- Cada entrada tiene un identificador único: su *Nombre distinguido* (*Distinguished Name*, DN). Este consta de su *Relative Distinguished Name* (RDN) construido por algunos atributos en la entrada, seguidos del DN de la entrada del padre. Pensar en el nombre distinguido como un completo nombre de archivo y el nombre distinguido relativo como el nombre de archivo relativo en un folder.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- ◆ **UTILIZACION DE OTROS ORÍGENES DE DATOS:**
- ◆ Hasta ahora hemos visto cómo obtener información de una base de datos, pero este no es el único método para almacenar información. También podemos almacenar información en ficheros por lo que resulta necesario ser capaces de trabajar con ellos.
- ◆ Los archivos **CSV** (del inglés *comma-separated values*) son un tipo de documento en [formato abierto](#) sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal: Argentina,México, Brasil...) y las filas por saltos de línea. Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles.
- ◆ Por otra parte los ficheros XML se utilizan hoy en día para el intercambio de información estructurada y está basado en un lenguaje de etiquetas

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

♦ UTILIZACION DE OTROS ORÍGENES DE DATOS:

- ♦ Para procesar un fichero, lo primero que tenemos que hacer es abrirlo o crearlo mediante una función.
Una vez abierto podremos tanto leer como escribir en él y una vez que hemos terminado de utilizarlo se cierra
- ♦ funciones que disponemos en PHP para interactuar con ficheros(las de ASP y JSP en el libro)
 - ♦ [fopen](#) — Abre un fichero o un URL
 - ♦ [fclose](#) — Cierra un puntero a un archivo abierto
 - ♦ [feof](#) — Comprueba si el puntero a un archivo está al final del archivo
 - ♦ [fgetcsv](#) — Obtiene una línea de un puntero a un fichero y la analiza en busca de campos CSV
 - ♦ [fputcsv](#) — Dar formato CSV a una línea y escribirla en un puntero a un fichero
 - ♦ [fwrite](#) — Escritura de un archivo en modo binario seguro
 - ♦ [fread](#) — Lectura de un fichero en modo binario seguro

Más informacin en <http://php.net/manual/es/ref.filesystem.php>

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

- ◆ **UTILIZACION DE OTROS ORÍGENES DE DATOS:**
- ◆ En PHP Y JSP, una de las alternativas para leer ficheros XML, es convertir este e un objeto DOM que permita recorrer la estructura en forma de árbol. Existen métodos para crear documentos XML y para acceder a as distintas partes que este posee:
 - ◆ `DOMDocument::createElement` — Crea un nuevo elemento
 - ◆ `DOMDocument::createAttribute` — Crear nuevo atributo
 - ◆ `DOMDocument::getElementsByTagNameNS` — Busca todos los elementos con el nombre de etiqueta dado en el espacio de nombre especificado.
 - ◆ `DOMElement::getAttribute` — Devuelve el valor de un atributo
 - ◆ `DomDocument:dump_file`

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ PROGRAMACION DE TRANSACCIONES

- ◆ Actualmente, utilizamos bases de datos que albergan a una gran numero de usuarios. Estos interactúan con la base de datos de manera simultánea, 1 o que puede provocar inconsistencias dentro de la misma, ya que si varios usuarios modifiquen de manera independiente la bases de datos la probabilidad de colisiones entre usuarios al actualizar los datos es bastante grande. Las transacciones surgieron como medida para controlar el acceso a los datos ante la necesidad de acceder de manera simultánea a la base de datos y el riesgo que esto acarrearba.
- ◆ Cuando hablamos de transacciones nos referimos a un conjunto de instrucciones que se ejecutan de manera atómica, es decir, como si fuera una sola instrucción o sentencia SQL. Gracias a la atmicidad que nos otorgan las transacciones podemos realizar operaciones complejas sobre la base de datos sin que por ello pongamos en peligro la integridad de la misma.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

◆ PROGRAMACION DE TRANSACCIONES

Como hemos dicho anteriormente, una transacción está compuesta por varias instrucciones que se comportan con una sola. Por ello, solo si todas y cada una de las instrucciones que la componen se ejecutan con éxito, esta lo hará también. De lo contrario, si alguna de las instrucciones falla se anulan todas las instrucciones que se hayan ejecutado hasta el momento para dejar la base de datos igual que estaba antes de que se empezara a ejecutar la transacción.

Para que un grupo de sentencias sean consideradas una transacción debe cumplir las siguientes propiedades del ACID:

- ◆ Atomicidad. El grupo de sentencias debe comportarse como un todo o ninguno. Si alguna de ellas no se pudiera ejecutar se debería poder deshacer o revertir
- ◆ Consistencia La base de datos debe quedarse en un estado consistente y coherente al finalizar la transacción
- ◆ Aislamiento Los datos modificados durante una transacción no deben ser visibles para otro usuario hasta que esta no finalice y se hayan realizado todos los cambios de manera definitiva
- ◆ Permanencia Los efectos de una transacción se almacenarán de forma permanente

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

🟢 PROGRAMACION DE TRANSACCIONES

Por defecto, las bases de datos se comportan como si cada sentencia fuera una transacción (transacciones implícitas). Esto quiere decir que cuando ejecutamos una sentencia que modifica los datos de una tabla (sentencias DML se almacena el resultado en el disco de forma automática. Para definir transacciones de manera explícita necesitamos escribir la sentencia que da comienzo a la transacción, que varia ligeramente dependiendo de la base de datos. Después del inicio de la transacción , incluiremos todas las sentencias SQL, que queremos que formen parte de esta. Por último finalizaremos la transacción

🟢 Existen dos formas de finalizar una transacción

- 🟢 Commit se ejecuta para finalizar la transacción de forma exitosa y hacer permanentes los cambios realizados durante la misma
- 🟢 Rollback se ejecutará para abortar la transacción porque algunas de las instrucciones que la componen se han ejecutado y han fallado. Esta instrucción deshace todos los cambios temporales que se pudieran haber hecho en la base de datos de manera que la base de datos quede igual que estaba antes de ejecutar la transacción

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Lenguajes de programación web en servidor

🟢 PROGRAMACION DE TRANSACCIONES

Sintaxis para definir una transacción en MySQL

```
START {TRANSACCION|BEGIN [WORK]}
```

```
Sentencia1_SQL
```

```
Sentencia2_SQL
```

```
....
```

```
ROLLBACK [WORK] [TO SAVEPOINT NOMBRE | TO punto_recuperacion]
```

```
COMMIT [WORK]
```

Dentro de una transacción podemos definir distintos puntos a través de los cuales volver atrás con un ROLLBACK TO sin deshacer toda la transacción. La sintaxis para definir estos puntos de retorno es la siguiente:

```
SAVEPOINT nombre
```

Para borrar puntos de recuperación creados en la transacción actual:

```
RELEASE SAVEPOINT punto_recuperacion
```

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Serialización o niveles de aislamiento

- ◆ Niveles de aislamiento: Distintos tipos de transacciones dependiendo del tipo de acceso que permitamos durante la ejecución de una transacción.
- ◆ Conceptos previos:
 - ◆ **Lectura sucia** Cuando T1 modifica los datos que lee T2 y luego se hace un rollback de T1. Por lo tanto T2 ha leído unos datos que no existen
 - ◆ **Lectura no repetible** En una transacción T1 queremos leer dos veces los mismos datos, pero entre ambas lecturas T2 elimina algunos datos. Esto provoca que tengamos resultados distintos en cada una de las lecturas, ya que en la segunda lectura nos faltarán datos con respecto a la lectura anterior.
 - ◆ **Lectura fantasma.** Es parecida a la anterior, ya que en una transacción T1 queremos leer dos veces los mismos datos. Esto provoca que al leer por segunda vez nos encontremos tuplas o listas ordenadas de elementos que anteriormente no existían.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Serialización o niveles de aislamiento

vamos a ver los distintos tipos de aislamiento y en cuales de ellos se producen los fenómenos anteriormente explicados:

- ◆ **Serializable.** Permite ver la base de datos antes o después de la ejecución de una transacción, pero nunca mientras se está realizando. Por lo tanto da la sensación de que las instrucciones se ejecutan unas detrás de otras. Esto solo es aplicable al usuario que lo elige
- ◆ **Read Committed.** Nivel que evita la lectura sucia permitiendo solo la lectura de los datos que han sido confirmados mediante la realización e un commit de otra transacción.
- ◆ **Repeatable read** Este nivel en cambio evita la lectura no repetible. Es igual que Read committed añadiéndole la restricción de que todo lo que veamos en una lectura inicial de una transacción debemos poder verlo en lecturas posteriores
- ◆ **Read Uncommitted** Si una transacción se ejecuta con este nivel de aislamiento puede ver las modificaciones realizadas por otra transacción, sean actualizaciones borrados o inserciones a pesar de no haber hecho un commit.

TEMA 4: LENGUAJES DE PROGRAMACION DE APLICACIONES WEB EN EL LADO SERVIDOR

Serialización o niveles de aislamiento

Por último vamos a ver cómo configurar los niveles de aislamiento en la base de datos:

MySQL

```
SET [SESSION|GLOBAL] TRANSACTION ISOLATION LEVEL READ UNCOMMITTED |  
READ COMMITTED | SERIALIZABLE | REPEATABLE READ
```

MySQL permite especificar si queremos que el nivel de aislamiento se aplique para la sesión actual (SESSION) o sea el que se use siempre desde ese momento (GLOBAL)