

Metodología Script

- Contenido
 - Filosofía Script
 - Diagrama Script
 - Fases de la metodología
 - Procesos Script



Metodología Script

- Contenido
 - **Filosofía Script**
 - Diagrama Script
 - Fases de la metodología
 - Procesos Script



Filosofía Script (1)

- La metodología Script o V-Script es una metodología de desarrollo de software que tiene un alto componente dinámico, orientado hacia la interfaz de usuario.
- Puede seguir un ciclo de vida clásico en V o en prototipado evolutivo.
- Se adapta perfectamente al paradigma de orientación a objetos, aunque se han usado técnicas Script en metodologías estructuradas, para el diseño de interfaz de usuario.
- Tiene la particularidad de dividir las etapas en dos grupos:
 - Etapas orientadas al cliente
 - Etapas orientadas a la computadora.



Filosofía Script (2)

- Mediante el proceso Script, se capturan las necesidades del usuario mediante la construcción de maquetas o prototipos desechables, tratando de capturar la expectativa del usuario: qué es lo que el usuario espera que haga el producto.
- A su vez, define las interfaces de usuario y permite integrar los aspectos del modelo estático y funcional.
- La metodología permite integrar perfectamente los procesos de estimación, gestión de calidad, gestión de configuración y verificación y validación del software.
- Esta metodología se adapta perfectamente al estándar IEEE 1074-1991: Standard for Developing Software Life Cycle Process.



Metodología Script

- Contenido
 - Filosofía Script
 - **Diagrama Script**
 - Fases de la metodología
 - Procesos Script



Diagrama Script (1)

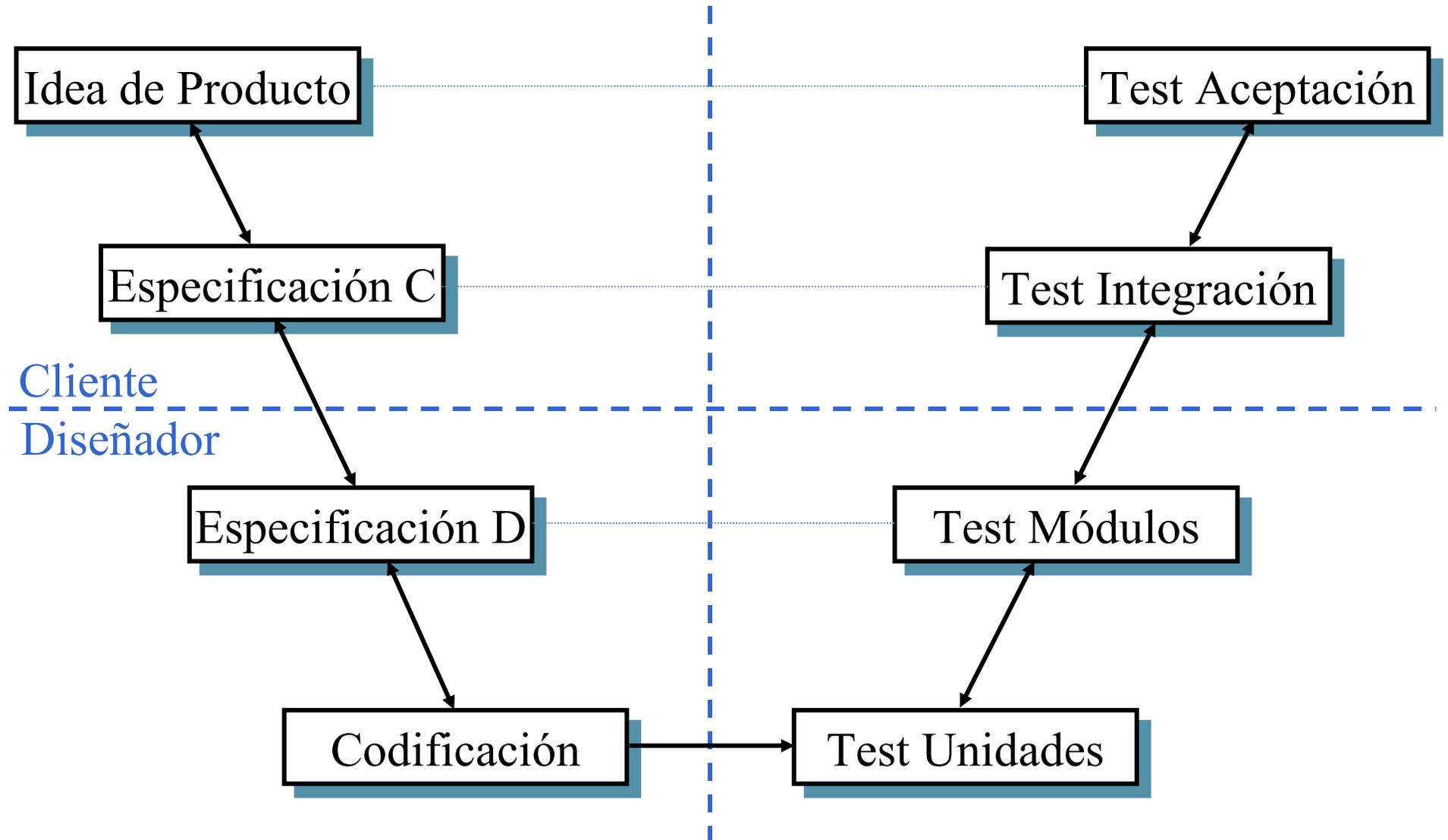


Diagrama Script (2)

- En el diagrama puede observarse dos ejes fundamentales:
 - **Eje horizontal:** divide al ciclo de vida en dos partes: una orientada hacia el cliente y otra orientada hacia el desarrollador computadora).
 - **Eje vertical:** divide al ciclo de vida en etapas de desarrollo y etapas de prueba del software.
- Después de finalizada cada etapa, se determina una **línea de base**. Una línea de base es un punto del ciclo de vida que permite tomar decisiones condicionantes. Se evalúa todo lo realizado hasta ese momento mediante revisiones formales y se decide seguir adelante (hacia la próxima línea base) o bien continuar en cada etapa.
- Las líneas base de esta metodología se encuentran al final de cada etapa Script.



Diagrama Script (3)

- Cada etapa Script de desarrollo tiene asociada una etapa de prueba al mismo nivel de abstracción. Estas etapas permiten verificar y validar el producto en los diferentes puntos del camino del producto desde el mundo real (necesidad) hacia la máquina.
- Dentro de cada etapa Script suelen encontrarse procesos con refinamiento sucesivo, que son limitados por las líneas base.
- Como el modelo V tradicional, puede volverse a la etapa anterior en caso de que se necesite realizar refinamiento.
- Dentro de un prototipado evolutivo, se recorren tantas veces el modelo como veces se necesita acercarse el prototipo al producto final.



Metodología Script

- Contenido
 - Filosofía Script
 - Diagrama Script
 - **Fases de la metodología**
 - Procesos Script



Fases de la Metodología

- Idea de Producto
- Especificación C
- Especificación D
- Codificación
- Test de Unidades
- Test de Módulos
- Test de Integración
- Test de Aceptación



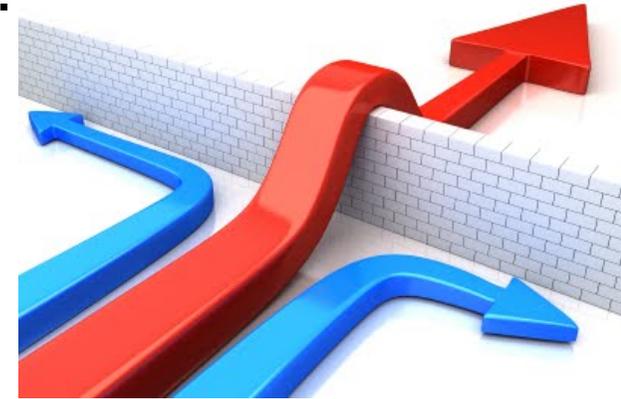
Fases de la Metodología

- **Idea de Producto**
- Especificación C
- Especificación D
- Codificación
- Test de Unidades
- Test de Módulos
- Test de Integración
- Test de Aceptación



Idea de Producto (1)

- Esta es la fase inicial de la metodología Script.
- **Documentos de Entrada:**
 - Surge el problema como una Necesidad.
- **Actividades:**
 - Identificación de usuarios participantes.
 - Planificación y Realización de Entrevistas (educación).
 - Análisis de Requisitos y Modelado de Casos de Uso.
 - Estudio de Viabilidad (Técnica, Operativa y Económica).
- **Documentos de Salida:**
 - Usuarios Participantes
 - Transcripción de Entrevistas y Documentación estudiada.
 - Modelo de Casos de Uso.
 - Estudio de Viabilidad.

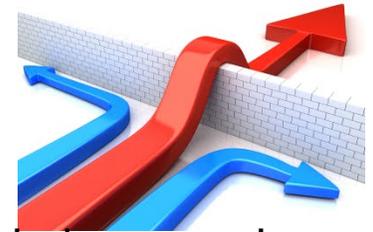


Idea de Producto (2)

- **Identificación de usuarios participantes:**

Los objetivos de esta tarea son identificar a los responsables de cada una de las unidades implicadas y a los principales usuarios implicados. Para ello se consideran los siguientes aspectos:

- Incorporación de usuarios al equipo de proyecto.
- Conocimiento de los usuarios de las funciones a automatizar.
- Repercusión del nuevo sistema sobre las actividades actuales de los usuarios.
- Implicaciones legales del nuevo sistema.



- **Planificación y Realización de Entrevistas:**

Esta tarea tiene como finalidad capturar los requisitos de usuarios para el desarrollo del sistema. El proceso comprende:

- Planificar las entrevistas a realizar: en la planificación se incluirá fecha, hora y lugar de la entrevista, duración estimada y guión de la entrevista.
- Realizar las entrevistas y documentarlas debidamente.
- Documentar los requisitos identificados con sus prioridades.

Idea de Producto (3)

- **Análisis de Requisitos y Modelado de Casos de Uso.**

Diagrama de Casos de Uso de Negocio, Descripciones Textuales de Casos de Uso, Diagramas de Actividad.

- **Estudio de Viabilidad (Técnica, Operativa y Económica).**

- **Viabilidad económica:** una evaluación del costo de desarrollo frente al beneficio final producido por el sistema desarrollado.
- **Viabilidad técnica:** un estudio de la funcionalidad, el rendimiento y las restricciones que pueden afectar a la posibilidad de realización de un sistema aceptable.
- **Viabilidad operativa (legal):** una determinación de cualquier infracción, violación o ilegalidad que pudiera resultar del desarrollo del sistema.

* (Pressman Cap. 5, pág 156)



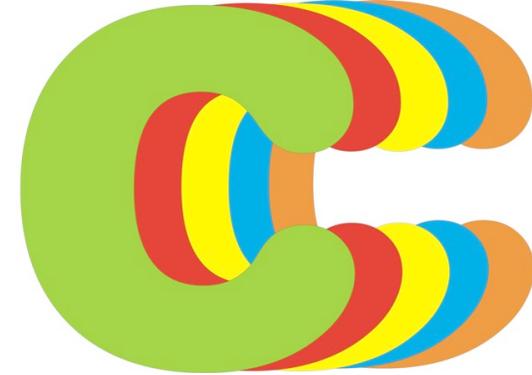
Fases de la Metodología

- Idea de Producto
- **Especificación C**
- Especificación D
- Codificación
- Test de Unidades
- Test de Módulos
- Test de Integración
- Test de Aceptación



Especificación C (1)

- Esta es la fase orientada al cliente
- **Documentos de Entrada:**
 - Son los documentos de salida de la Idea de Producto.
- **Actividades:**
 - Modelización del Sistema.
 - Identificación de Roles
 - Diseño de guiones y escenarios y creación de maqueta inicial.
 - Sesiones con usuario. Refinamiento de guiones y escenarios.
 - Construcción de diagramas y tablas de transición de escenarios
 - Descripción de escenarios y objetos de escenario.
- **Documentos de Salida:**
 - Modelos del Sistema (Clases, Transición de Estados, Interacción)
 - Identificación de Roles.
 - Documentación de guiones.
 - Documentación de Escenarios y Objetos de Escenario.



Especificación C (2)

- **Modelización del Sistema:**

Se modeliza el sistema utilizando las técnicas de análisis orientado a objetos ya estudiadas (Modelo de Clases, Diagramas de Interacción de Casos de Uso y Diagramas de Transición de Estados).

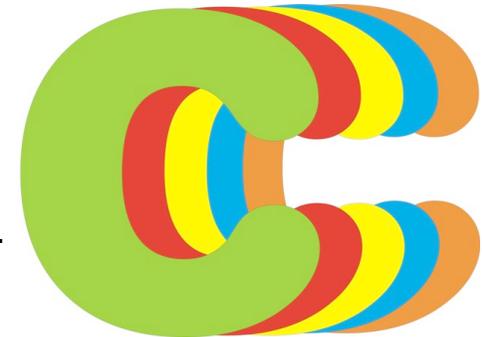
- **Identificación de Roles:**

Se trata de identificar los roles que cumplen los usuarios finales en el sistema, teniendo en cuenta las funciones que realizan y describiendo las restricciones a los datos de cada uno. Se elabora una lista de cada rol identificado con las funciones que realiza y las restricciones de datos.



Especificación C (3)

- **Diseño de guiones, escenarios y creación de maqueta inicial:**
 - Se deben construir los guiones y escenarios iniciales para la captura de requerimientos de usuario.
 - Hay que tener en cuenta el entorno tecnológico, es decir las plataformas de software y hardware donde se ejecuta el sistema.
 - Los guiones se construyen mediante diagramas de transición de escenarios.
 - Los escenarios se construyen de acuerdo a la funcionalidad deseada, identificando los objetos del escenario que lo componen.
 - Los criterios a tener en cuenta son los siguientes:
 - Para comenzar, construir un escenario por cada función soportada, agregando un escenario del menú principal.
 - Construir interfaces lo más sencillas y amigables posibles.
 - Definir estándares para tipos de letra, tamaños, posiciones, etc.
 - Pedir información con una secuencia lógica.
 - Permitir al usuario cancelar parte de la transacción o cancelarla toda.
 - Proporcionar alternativas por omisión para las entradas estándar.
 - Aprovechar el color y el sonido, pero sin abusar.
 - Con los guiones y escenarios se construyen las maquetas para capturar los requerimientos del usuario, aprovechando la rapidez y precisión de los lenguajes de cuarta generación.



Especificación C (4)

- **Sesiones con usuario. Refinamiento de guiones y escenarios:**
 - Usando la maqueta inicial y en sesiones con el usuario, se captura la expectativa de éste, es decir, lo que el usuario espera que haga cada escenario.
 - Se van modificando los guiones y escenarios agregando la descripción funcional que el usuario hace de cada escenario y los eventos que producen las transiciones entre los mismos.
 - Se adoptan y modifican estándares de objetos de entorno.
 - Al final de cada sesión, se documentan los guiones y escenarios con su descripción funcional y se modifica la maqueta.
 - Se realizan nuevas sesiones refinando los escenarios hasta que se satisfagan las expectativas del usuario y a continuación se proceden a validar los guiones y escenarios finales.



Especificación C (5)

- **Construcción de diagramas y tablas de transición de escenarios:**



- La documentación de guiones consiste en la especificación de los diagramas de transición de escenarios y las tablas de transición de escenarios.
- Las tablas de transición de escenarios muestran los eventos que producen las transiciones entre escenarios y la secuencia de los mismos, ayudando a detectar y corregir errores.

- **Descripción de escenarios y objetos de escenario:**

- En este apartado se deben especificar los escenarios y los objetos que lo componen.
- Cada escenario debe ir acompañado con una descripción funcional.
- A continuación de cada diagrama de escenario se incluyen las fichas de objeto de escenario que capturan los atributos de cada objeto de escenario.

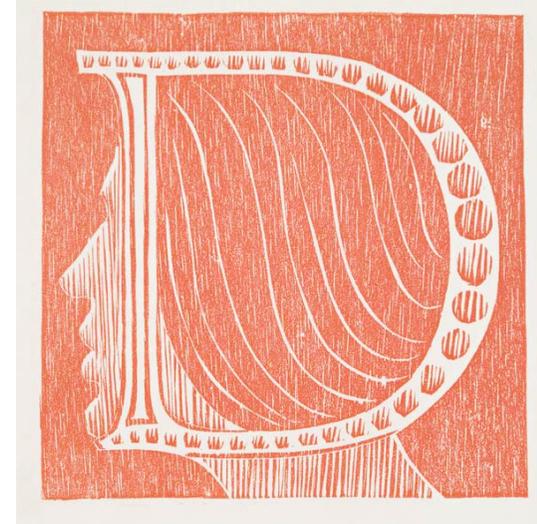
Fases de la Metodología

- Idea de Producto
- Especificación C
- **Especificación D**
- Codificación
- Test de Unidades
- Test de Módulos
- Test de Integración
- Test de Aceptación



Especificación D (1)

- Esta es la fase orientada al desarrollador (computadora)
- **Documentos de Entrada:**
 - Son los documentos de salida de la Especificación C.
- **Actividades:**
 - Elección de Plataformas.
 - Diseño de la Arquitectura Física del Sistema.
 - Fichas Técnicas de Métodos de Objetos.
 - Fichas Técnicas de Eventos de Objetos de Escenario.
 - Fichas Técnicas de Procedimientos Asociados.
 - Diseño del Modelo Lógico y Físico de Datos del Sistema.
- **Documentos de Salida:**
 - Plataformas y Arquitectura Física del Sistema.
 - Fichas Técnicas de Métodos de Objetos, Eventos de Objetos de Escenario y Procedimientos Asociados.
 - Modelos Lógico y Físico de Datos del Sistema.



Especificación D (2)

- **Elección de Plataformas:**

En esta etapa se eligen las plataformas de software y hardware así como también el entorno de desarrollo del producto.

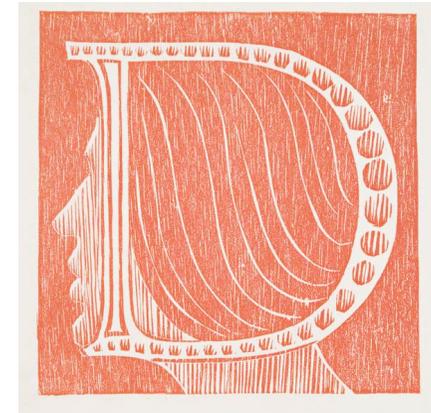
- **Diseño de la Arquitectura Física del Sistema:**

Se pretende obtener la estructura modular y los detalles de proceso del sistema. Se utilizan los siguientes modelos:

- Modelo Servidor de Archivos.
- Modelo Cliente/Servidor de 2 capas.
- Modelo Cliente/Servidor de 3 capas.

- **Fichas Técnicas de Métodos de Objetos:**

Se elaboran las fichas de los métodos de las clases del modelo de clase usando diagramas de flujo o pseudocódigo.



Especificación D (3)

- **Fichas Técnicas de Eventos de Objetos de Escenario:**



Se identifican los eventos sobre los objetos de entorno que disparan procedimientos y se describen los mismos en una ficha de eventos o ficha técnica de servicios de objeto utilizando diagramas de flujo o pseudocódigo.

- **Fichas Técnicas de Procedimientos Asociados:**

Son aquellos procedimientos que codifican reglas de negocio y son utilizados del lado del servidor generalmente.

- **Diseño del Modelo Lógico y Físico de Datos del Sistema:**

- Se elaboran el modelo lógico (generalmente se elige el modelo relacional) y el modelo físico del sistema (incluyendo los tipos de datos e índices teniendo en cuenta el Sistema Gestor de Base de Datos)

Fases de la Metodología

- Idea de Producto
- Especificación C
- Especificación D
- **Codificación**
- Test de Unidades
- Test de Módulos
- Test de Integración
- Test de Aceptación



Codificación (1)

- Fase de elaboración del código fuente y ejecutables de la aplicación.
- **Documentos de Entrada:**
 - Son los documentos de salida de la Especificación D.
- **Actividades:**
 - Elección de el/los lenguaje(s).
 - Escritura del código.
 - Documentación del código.
 - Compilación.
- **Documentos de Salida:**
 - Código Fuente Documentado.
 - Archivos compilados de la aplicación.



Codificación (2)

- **Elección de el/los lenguaje(s):**
 - Los criterios que se aplican durante la evaluación de los lenguajes son:
 - Área de aplicación general.
 - Complejidad algorítmica y computacional.
 - Entorno en el que se ejecutará el software.
 - Consideraciones de rendimiento.
 - Complejidad de las estructuras de datos.
 - Conocimiento de la plantilla de desarrollo de software.
 - Independientemente del paradigma de ingeniería del software, el lenguaje de programación tendrá impacto en la planificación, el análisis, el diseño, la codificación, la prueba y el mantenimiento de un proyecto.



Codificación (3)

- **Escritura del código:**

- Una vez que se genera el código fuente, la función de un módulo debe resultar clara sin necesidad de referirse a ninguna especificación del diseño.
- El estilo de codificación conlleva una filosofía de codificación que mezcle la simplicidad con la claridad.
- La forma en que el código fuente aparece en el listado es una importante contribución a la legibilidad. El sangrado del código fuente realiza las construcciones lógicas y los bloques de código tabulando desde el margen izquierdo de forma que se vean desplazados esos atributos.



Codificación (4)

- **Documentación del código:**

- La documentación interna del código fuente comienza con la elección de los nombres de los identificadores (variables y etiquetas), continúa con la localización y la composición de los comentarios y termina con la organización visual del programa. La elección de nombres de identificadores significativos es crucial para la legibilidad.
- El lenguaje debe permitir la posibilidad de expresar comentarios en lenguaje natural como parte del listado del código fuente. Con unos recursos mnemotécnicos apropiados para los identificadores y unos buenos comentarios, se asegura una documentación interna adecuada.



Fases de la Metodología

- Idea de Producto
- Especificación C
- Especificación D
- Codificación
- **Test de Unidades**
- Test de Módulos
- Test de Integración
- Test de Aceptación



Test de Unidades (1)

- Prueba de las unidades individuales de código.
- **Documentos de Entrada:**
 - Son los documentos de salida de la Codificación.
- **Actividades:**
 - Pruebas de Caja Blanca.
- **Documentos de Salida:**
 - Resultados de la prueba de caja blanca.



Test de Unidades (2)

- **Pruebas de Caja Blanca:**

Es un método de prueba que permite detectar los siguientes errores internos del código de cada módulo. Se obtienen casos de prueba que:

- Garantizan que se ejercitan por lo menos una vez todos los caminos independientes de cada módulo.
- Ejercitan todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecutan todos los bucles en sus límites operacionales.
- Ejercitan las estructuras internas de datos para asegurar su validez.



Se utilizan técnicas de grafo de flujo (Pressman pág 630) y su complejidad ciclomática. Hay que tener en cuenta que “los errores se esconden en los rincones y se aglomeran en los límites”.

Fases de la Metodología

- Idea de Producto
- Especificación C
- Especificación D
- Codificación
- Test de Unidades
- **Test de Módulos**
- Test de Integración
- Test de Aceptación



Test de Módulos (1)

- Prueba de los módulos funcionales del sistema.
- **Documentos de Entrada:**
 - Son los documentos de salida de la Codificación.
- **Actividades:**
 - Pruebas de Caja Negra.
 - Pruebas de Estrés.
- **Documentos de Salida:**
 - Resultados de la prueba de caja negra.
 - Resultados de la prueba de estrés.



Test de Módulos (2)

- **Pruebas de Caja Negra:**

Se ve a cada módulo como una caja negra y se generan conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales del programa, observando las salidas. Se detectan los siguientes errores:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.



La prueba de la caja centra su atención en la información y la clave está en generar el conjunto de datos o condiciones de entrada.

- **Pruebas de Estrés:**

Se centra en realizar el análisis de valores límite, y en condiciones límite, ya que se ha demostrado que los errores tienden a darse más en los límites del campo de entrada y sometidos a condiciones límite.

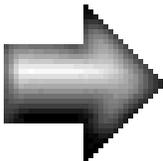
Fases de la Metodología

- Idea de Producto
- Especificación C
- Especificación D
- Codificación
- Test de Unidades
- Test de Módulos
- **Test de Integración**
- **Test de Aceptación**



Test de Integración (1)

- Prueba de la estructura modular del programa y su interacción.
- **Documentos de Entrada:**
 - Son los documentos de salida de la Codificación.
- **Actividades:**
 - Pruebas de Integración
- **Documentos de Salida:**
 - Resultados de la prueba de integración.



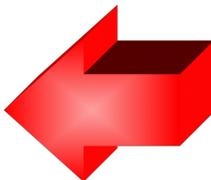
Test de Integración (2)

- **Pruebas de Integración:**

Los errores que surgen de integrar los módulos son:



- Los datos se pueden perder en una interfaz: un módulo puede tener un efecto adverso e inadvertido sobre otro.
- Las subfunciones, cuando se combinan, pueden no producir la función principal.
- Las estructuras de datos globales pueden presentar problemas.
- El objetivo es tomar los módulos probados y construir una estructura de programa que esté de acuerdo con lo que dicta la Especificación C.
- Existen dos tipos de integración:
 - Integración descendente: se integran los módulos moviéndose hacia abajo por la jerarquía de control, comenzando con el módulo de control principal.
 - Integración ascendente: se integran los módulos atómicos (niveles más bajos) primero y luego se continúa con el nivel inmediato superior.



Fases de la Metodología

- Idea de Producto
- Especificación C
- Especificación D
- Codificación
- Test de Unidades
- Test de Módulos
- Test de Integración
- **Test de Aceptación**



Test de Aceptación (1)

- Prueba del software para validar si funciona de acuerdo con las expectativas razonables del cliente.
- **Documentos de Entrada:**
 - Son los documentos de salida de la Codificación.
- **Actividades:**
 - Pruebas α y β .
- **Documentos de Salida:**
 - Resultados de las pruebas α y β .



Test de Aceptación (2)

- **Pruebas α y β :**

- La prueba α es conducida por el cliente en el lugar de desarrollo. Se usa el software de forma natural (previa capacitación), con el encargado de desarrollo mirando “por encima del hombro” del usuario y registrando errores y problemas de uso. Se llevan a cabo en un entorno controlado.
- La prueba β se lleva a cabo en uno o más lugares de clientes, por los usuarios finales de software. El encargado de desarrollo no está presente. El cliente registra todos los problemas (reales o imaginarios) que encuentra durante la prueba e informa a intervalos regulares al equipo de desarrollo.
- Tanto los planes como los procedimientos de prueba, estarán diseñados para asegurar que se satisfacen todos los requisitos funcionales y que se alcanzan todos los requisitos de rendimiento.



Metodología Script

- Contenido
 - Filosofía Script
 - Diagrama Script
 - Fases de la metodología
 - **Procesos Script**



Procesos Script

- La Especificación C
 - El proceso Script propiamente dicho
 - Identificación de Roles
 - Guiones y Escenarios
- La Especificación D
 - Diseño Estructural
 - Diseño Detallado
- Diseño de Datos
 - Modelo Lógico y Físico de Datos
- Codificación
 - Documentación del Código



Procesos Script

- La Especificación C
 - **El proceso Script propiamente dicho**
 - Identificación de Roles
 - Guiones y Escenarios
- La Especificación D
 - Diseño Estructural
 - Diseño Detallado
- Diseño de Datos
 - Modelo Lógico y Físico de Datos
- Codificación
 - Documentación del Código



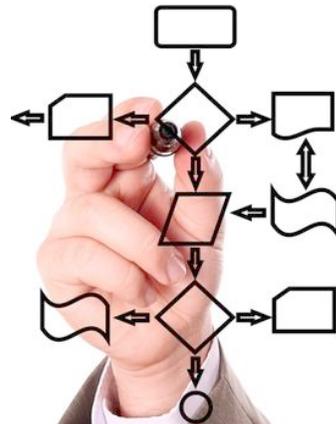
El Proceso Script propiamente dicho (1)

- El diseño de las aplicaciones es fuertemente dependiente de la interfaz de usuario y requiere una especial atención ya que en los tiempos actuales el éxito o fracaso de un sistema está sujeto a la estética, funcionalidad, simplicidad de la aplicación, características presentes en una buena interfaz de usuario.
- Para definir el aspecto interactivo y capturar con éxito todas estas características del sistema en su relación con el usuario, se especifican las interfaces de usuario, también llamadas escenarios en donde se desarrolla la interacción, usando la **técnica de guiones**.
- Esta técnica tiene en cuenta la **expectativa** y **necesidades** que tiene el usuario en cada escenario para realizar las funciones del sistema y la secuencia en que se suceden dichos escenarios.
- Los escenarios y sus transiciones forman lo que se conoce como **guión**. Se consideran **escenarios** las ventanas, las páginas Web, los diálogos e informes.

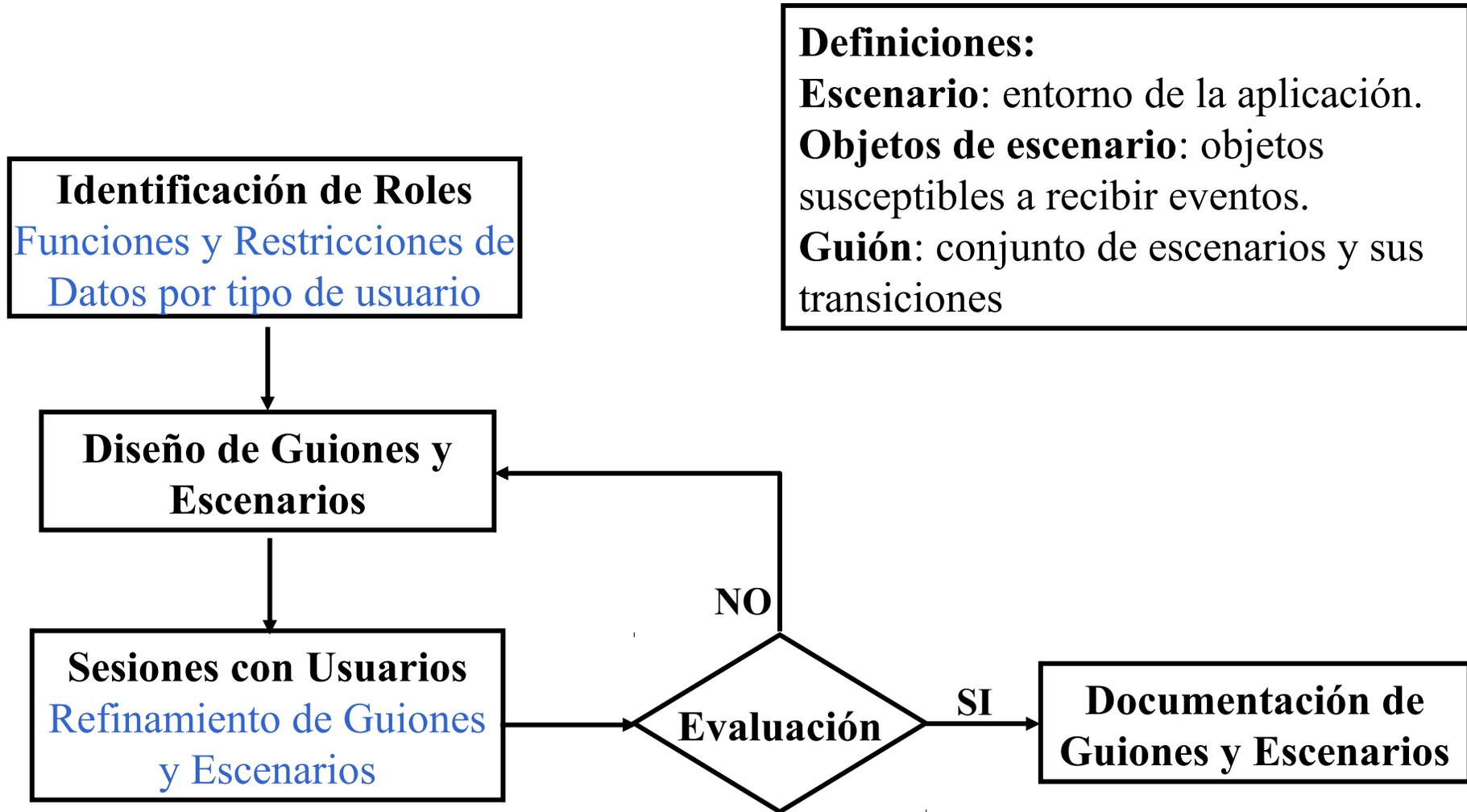


El Proceso Script propiamente dicho (2)

- Los escenarios están formados a su vez por **objetos de escenario** (botones, cuadros de texto, listas, etc.) susceptibles a recibir eventos.
- Los **eventos** sobre los objetos de entorno responden a los estándares del sistema operativo, y disparan procedimientos cuyas características dependerán de la expectativa del usuario.
- La descripción de los guiones y escenarios se obtiene sobre la base de sesiones con el usuario usando **prototipos desechables (maquetas)**.
- Estas descripciones siguen la expectativa del usuario en cuanto a la respuesta de los objetos de entorno a los eventos que se le aplican.



El Proceso Script propiamente dicho (3)



Procesos Script

- La Especificación C
 - El proceso Script propiamente dicho
 - **Identificación de Roles**
 - Guiones y Escenarios
- La Especificación D
 - Diseño Estructural
 - Diseño Detallado
- Diseño de Datos
 - Modelo Lógico y Físico de Datos
- Codificación
 - Documentación del Código



Identificación de Roles

- Identificación de Roles:

- Rol Encargado de Biblioteca

- Funciones:

- Altas, bajas, modificaciones y listado de libros.
 - Altas, bajas, modificaciones y listado de lectores.
 - Gestión de préstamos y devoluciones de libros.
 - Consulta sobre disponibilidad de libros.
 - Informe de préstamos y devoluciones.

- Restricciones de datos:

- No tiene

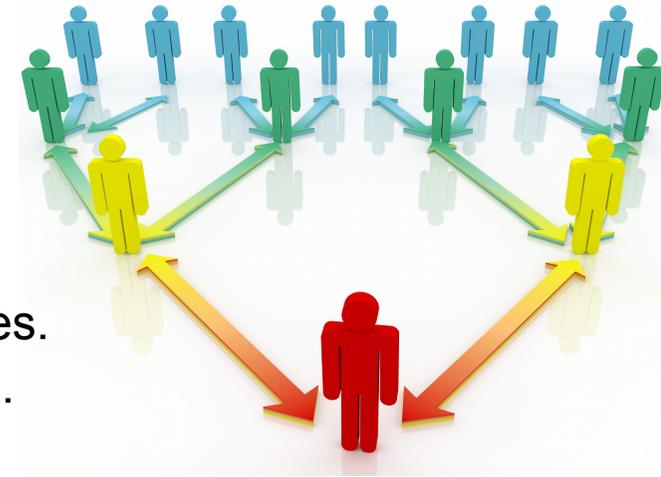
- Rol Lector:

- Funciones:

- Consulta sobre disponibilidad de libros.

- Restricciones de datos:

- No puede modificar información sobre libros.
 - No puede modificar información sobre lectores.
 - No puede modificar información sobre préstamos y devoluciones.



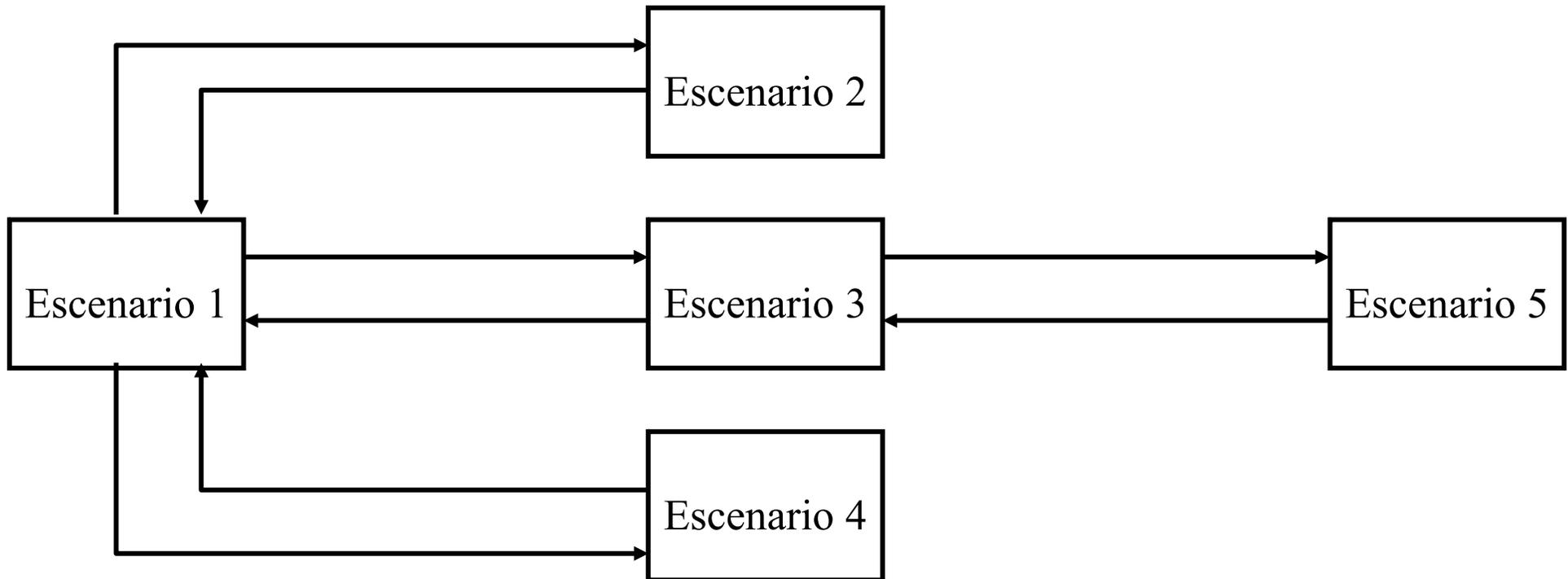
Procesos Script

- La Especificación C
 - El proceso Script propiamente dicho
 - Identificación de Roles
 - **Guiones y Escenarios**
- La Especificación D
 - Diseño Estructural
 - Diseño Detallado
- Diseño de Datos
 - Modelo Lógico y Físico de Datos
- Codificación
 - Documentación del Código



Guiones y Escenarios (1)

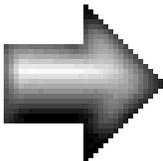
- La documentación de guiones consiste en la especificación de los **diagramas de transición de escenarios** y las **tablas de transición de escenarios**.



Guiones y Escenarios (2)

- Las tablas de transición de escenarios muestran los eventos que producen las transiciones entre escenarios y la secuencia de los mismo, ayudando a detectar y corregir errores.

	Escenario 1	Escenario 2	Escenario n
Escenario 1		Evento 1		Evento m
Escenario 2	Evento 2			
.....				
Escenario n	Evento 3			



Guiones y Escenarios (3)

- Los controles que deben realizarse son:
 - Comprobar que todos los escenarios tienen por lo menos un evento que conduce a otro escenario (fila no vacía).
 - Comprobar que todos los escenarios son llamados al menos por un evento de otro escenario (columna no vacía).
 - Podrá existir el caso de un escenario que no sea llamado por ningún otro, en cuyo caso debe coincidir con el escenario de inicio de la aplicación.



Guiones y Escenarios (4)

- Se deben especificar los escenarios y los objetos que lo componen. Cada escenario debe ir acompañado con una descripción funcional.
- La nomenclatura de identificación de escenarios y objetos se describe a continuación:
 - Dos letras que identifican al usuario al que pertenece el escenario; luego siguen dos números que identifican al escenario; finalmente dos números que identifican al objeto del escenario.
 - Así el objeto SE_09_21, será el objeto 21 del escenario 09 del guión secretaría. El objeto 00 se reserva para el escenario propiamente dicho, que también es susceptible de interactuar con el usuario en forma de eventos.



Guiones y Escenarios (5)

Escenario SE_02_00: Mantenimiento de códigos postales

Este escenario permite realizar las altas, bajas y modificaciones de los códigos postales. Al cargarse la ventana se muestra el primer registro y si no existen registros todavía se muestra la planilla lista para cargar uno nuevo. Los botones permiten mover al primer registro(10), al registro anterior al que se muestra (11), al registro siguiente al que se muestra (12) y al último registro (13). Los botones de función permiten crear un registro nuevo (14), almacenar o modificar el existente (15) y borrar el registro que se muestra (17). El botón 04 permite visualizar una lista de códigos postales existentes ordenados por CP de manera de poder elegir uno para su tratamiento.

01

02 Código Postal: 1000 03

04 Buscar...

05 Provincia: Buenos Aires 06

07 Ciudad: Capital Federal 08

09

10 << 11 < 12 > 13 >>

14 15 16 17 Salir

Guiones y Escenarios (6)

- Para la descripción de los objetos de entorno se utilizan fichas de objetos de escenario:

	Nombre	Objeto 1
	Descripción	
Valores iniciales	Tipo de letra	
	Color	
	Tamaño	
	

Guiones y Escenarios (7)

- Para la descripción de los objetos de entorno se adoptan estándares de tipos de objetos con similares características para facilitar la descripción. Estos estándares contienen las propiedades y los valores comunes a todos los objetos. Luego en la descripción de los mismos se procede a agregar las propiedades particulares de cada objeto.

- Ejemplo:** Estándar de Ventanas

- Estilo borde = 1 'No permitir cambiar tamaño
- Tipo de letra
 - Nombre = "Arial"
 - Tamaño = 10
 - Subrayado = 0 'Falso
 - Itálica = 0 'Falso
- Botón maximizar = 0 'Falso
- Botón minimizar = 0 'Falso
- Permitir movimiento = 0 'Falso



Procesos Script

- La Especificación C
 - El proceso Script propiamente dicho
 - Identificación de Roles
 - Guiones y Escenarios
- La Especificación D
 - **Diseño Estructural**
 - Diseño Detallado
- Diseño de Datos
 - Modelo Lógico y Físico de Datos
- Codificación
 - Documentación del Código



Diseño Estructural (1)

- El objetivo de esta actividad es obtener la estructura modular y los detalles de proceso del sistema, partiendo de los productos obtenidos de la Especificación C.
- Para ello se utiliza una arquitectura cliente/servidor en un modelo de n capas.
- Cliente-servidor se refiere a un proceso que incluye al menos a dos entidades independientes, una el cliente y otra el servidor.
- El cliente realiza el requerimiento al servidor y el servidor cumple su proceso.
- Estas entidades pueden estar separadas y la lógica de la aplicación distribuida entre el cliente y el servidor.

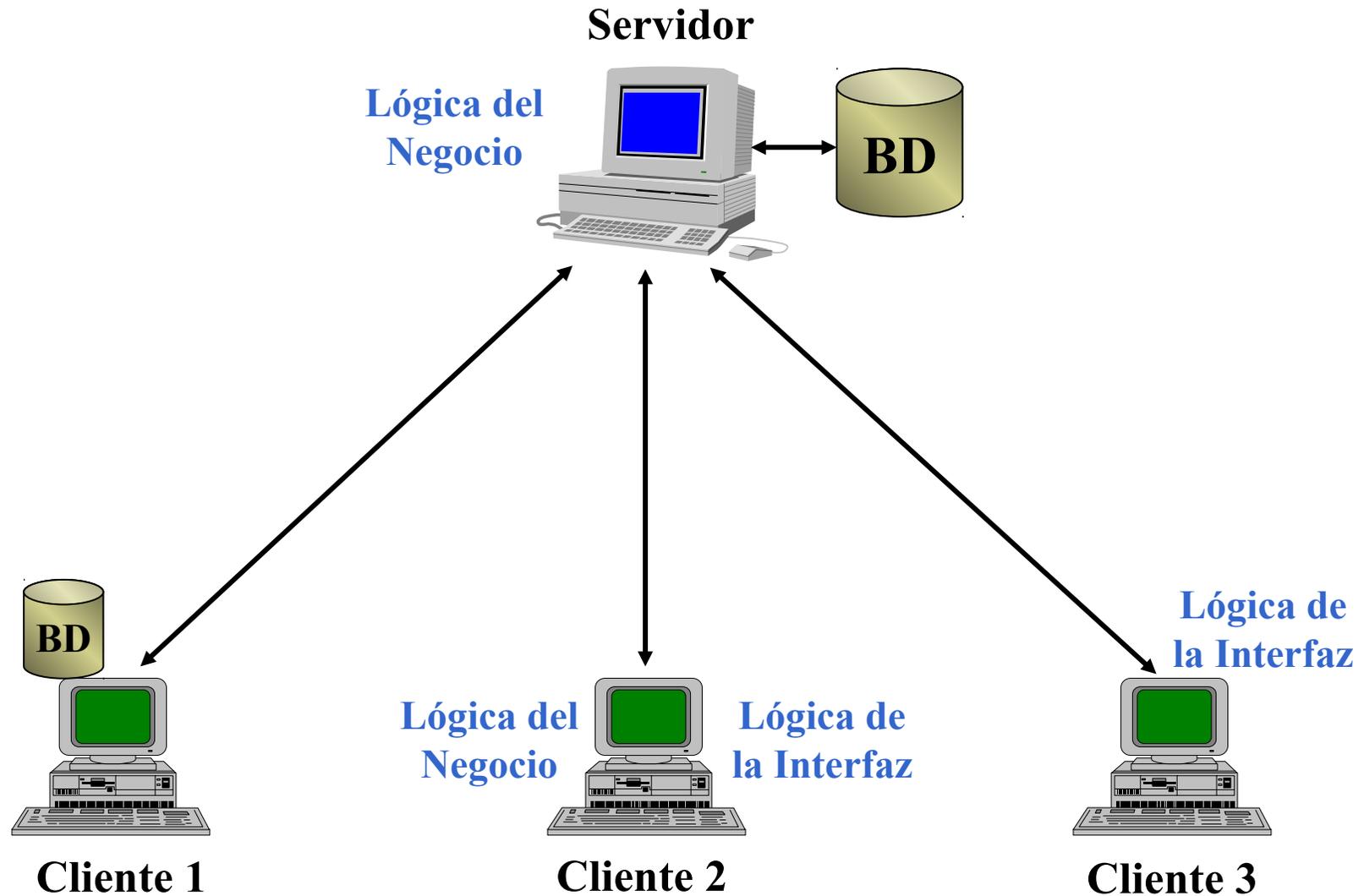


Diseño Estructural (2)

- El modelo cliente/servidor de dos capas tiene relación a la implementación física: uno o varios ordenadores de sobremesa operando como clientes y un servidor de red que alberga un motor de base de datos.
- En este modelo la lógica se divide entre estas dos posiciones físicas: el cliente y el servidor.
- La parte frontal del sistema (interfaz con el usuario) reside en los clientes y la lógica del negocio de la aplicación debe residir físicamente en el cliente o debe estar implementada en el servidor, dentro del sistema de gestión de base de datos.



Diseño Estructural (3)



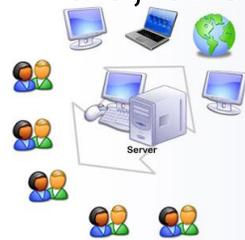
Diseño Estructural (4)

- Las ventajas de este modelo son:
 - Bajo costo inicial y flexibilidad de las implementaciones (permite ampliar y reducir los recursos informáticos según las necesidades).
 - Simplifica el acceso a datos, lo cual permite el desarrollo rápido de aplicaciones.
 - Brinda una estructura ampliable según las necesidades cambiantes del negocio, facilitando el mantenimiento del sistema y abaratando costes.
 - Brinda mejoras de rendimiento para la aplicación ya que los requerimientos se ejecutan en el servidor, por lo que se reduce el tráfico en la red y se consigue un tiempo de ejecución increíblemente más corto que en un servidor de archivos.



Diseño Estructural (5)

- Para el desarrollo del modelo es necesario definir las interfaces con el usuario en cada uno de los clientes y la lógica del negocio, obtenida en el análisis del sistema, que residirá parte en los clientes y parte en el servidor, dentro del sistema gestor de base de datos.
- Las interfaces fueron definidas en el análisis del sistema mediante la técnica de guiones.
- Se obtienen los escenarios y su funcionalidad mediante los objetos de escenario.
- Los escenarios y los eventos sobre objetos de escenario deben residir en el cliente, mientras que los servicios solicitados al servidor de base de datos deben ser codificados en procedimientos o triggers en el servidor.
- Cada vez que se necesite un servicio, el evento lo convoca parametrizando el procedimiento.



Procesos Script

- La Especificación C
 - El proceso Script propiamente dicho
 - Identificación de Roles
 - Guiones y Escenarios
- La Especificación D
 - Diseño Estructural
 - **Diseño Detallado**
- Diseño de Datos
 - Modelo Lógico y Físico de Datos
- Codificación
 - Documentación del Código



Diseño Detallado (1)

- Se debe tener en cuenta que los eventos sobre los objetos de entorno responden a los estándares del Sistema operativo, y disparan procedimientos que son de carácter local a la ventana donde se producen.
- Eso no impide que modifiquen variables globales, ni convoquen procedimientos globales, ni soliciten servicios remotos del tipo de esquema cliente-servidor.
- La ventana pasa el control al procedimiento correspondiente al producirse el evento que, una vez concluido, devuelve el control a la ventana.
- Algunas veces los procedimientos convocan a otras ventanas o disparan eventos de la misma ventana, en cuyo caso se interrumpe la ejecución del procedimiento, se pasa el control a la ventana o evento correspondiente y luego de ejecutarse el/los procedimiento(s), vuelve el control al procedimiento que los convocó. Al descargarse una ventana, devuelve el control al procedimiento que la convocó.



Diseño Detallado (2)

- Se identifican los eventos sobre los objetos de entorno que disparan procedimientos y se describen los mismos en una ficha técnica de eventos sobre objetos de escenario.
- Para la representación de los procedimientos se pueden usar diferentes técnicas como diagramas de flujo, pseudocódigo, etc.

Nombre:	Evento 1	Evento	(tipo de evento)
(procedimiento)			



Diseño Detallado (3)

Mantenimiento de Profesores

Código Profesor: B006

Apellidos: Abdala

Nombres: Javier

Domicilio: San Juan 623

Código Postal: 4000

Provincia: Tucumán

Ciudad: San Miguel de Tucumán

Teléfono: 217664

<< < > >> Salir

Nombre:	SE_04_26	Evento	Click
Si SE_04_03="" o SE_04_06 = "" o SE_04_08 = "" o SE_04_10 = "" o SE_04_12 = "" entonces Mensaje("Datos incorrectos") Sino Buscar_registro (Profesores, "Código_profesor=SE_04_03", registro, éxito) Si NO éxito entonces Profesores.Contraseña_profesor=SE_04_03 Nuevo(Profesores, datos de planilla) Sino Actualizar(Profesores, registro, datos de planilla)			

Procesos Script

- La Especificación C
 - El proceso Script propiamente dicho
 - Identificación de Roles
 - Guiones y Escenarios
- La Especificación D
 - Diseño Estructural
 - Diseño Detallado
- Diseño de Datos
 - **Modelo Lógico y Físico de Datos**
- Codificación
 - Documentación del Código



Procesos Script

- La Especificación C
 - El proceso Script propiamente dicho
 - Identificación de Roles
 - Guiones y Escenarios
- La Especificación D
 - Diseño Estructural
 - Diseño Detallado
- Diseño de Datos
 - Modelo Lógico y Físico de Datos
- Codificación
 - **Documentación del Código**



Documentación del Código

```
Public CX_int As String
Public Nombre_int, Apellido_int, Contraseña_int,
    Carrera_int As String
Public UE_int As Integer
Public Promedio_int As Single

Public Muestra_camino, Muestra_titulo, Muestra_mensaje
As String

Public Lista_int(1 To 20) As String
Public n_int As Integer

Private Sub Form_KeyPress(KeyAscii As Integer)
    clave = clave & Chr(KeyAscii)
    If Len(clave) = 7 Then
        'Veo si es tarjeta de salida
        If clave = "0000000" Then
            Unload Me
            Exit Sub
        End If
        'Verifica la clave en código_alumno de la base de datos
        Alumnos.FindFirst "CX=" & clave & ""
        'Si encuentro alumno
        If Not Alumnos.NoMatch Then
            'Transfero los datos a las variables de intercambio
            CX_int = clave
            Nombre_int = Alumnos("Nombre_alumno")
            Apellido_int = Alumnos("Apellido_alumno")
            Contraseña_int = Alumnos("Contraseña_alumno")
            UE_int = Alumnos("unidades_estudio_alumno")
            Promedio_int = Alumnos("promedio")
            Carrera_int = Alumnos("carrera")
            'Descargo y muestro ventana de contraseña
            Unload Me
            AL_02_00.Show 1
        Else
            MsgBox ("Esta tarjeta no es de ningún alumno
registrado en el sistema. Dé aviso a Secretaría")
            clave = ""
        End If
    End If
End Sub
```

Metodología Script

- Contenido
 - Filosofía Script
 - Diagrama Script
 - Fases de la metodología
 - Procesos Script

