TEMA 3: EL MARCO DE CONTROL DE APLICACION

“Los controles de aplicación buscan garantizar que las aplicaciones cumplan los objetivos de AI”.

En este tema se verán en detalle los siete grandes grupos de controles de aplicación en su orden natural de secuencia, desde que los datos entran en el sistema (desde el origen de los datos), continuando con su proceso y posterior salida y finalizando con su almacenamiento.

Estos grupos de controles son los siguientes:

1. Controles de captura, preparación y entrada de datos
2. Controles de acceso y comunicaciones
3. Controles de entrada
4. Controles de proceso
5. Controles de salida
6. Controles de pistas de Auditoria
7. Controles de copias de seguridad y recuperación.

3.1 Controles de Captura, Preparación y Entrada de datos

La introducción de datos en un ordenador consta de tres pasos: La captura de datos (proceso de identificar y recoger datos del mundo real, relevantes a los procesos a realizar), la preparación de datos (proceso de convertir los datos capturados en un formato entendible por el ordenador) y la entrada de datos (proceso de lectura de los datos para su introducción en el ordenador)

Las dos primeras fases son muy importantes para el auditor, ya que conllevan procesos monótonos que por tanto son propensos al error, ya que precisan intervención humana para su realización.

Con respecto a la última fase, en la actualidad existen controles hardware y software que garantizan que la entrada sea la correcta y por ello, normalmente no interesan tanto al auditor.

3.1.1 Evaluación de los métodos de Captura de datos.

Hace algunos años, la captura de datos estaba basada casi totalmente en la utilización de documentos de captura, y se caracterizaba por ser simple y flexible, requiriendo un gran esfuerzo y mucha intervención humana.

En la actualidad, la entrada directa de datos (con la utilización de ordenadores personales, o aplicaciones de captura de datos) reduce la intervención humana y por tanto los costes, pero requiere una mayor formación de los usuarios, y presenta un coste de hardware y software adicional, para los determinados puntos de captura de datos.

No obstante, se siguen existiendo métodos híbridos de captura de datos, mezcla de los dos anteriores, y que por tanto presentan asimismo una mezcla de sus ventajas e inconvenientes.

Control	Naturaleza	Aplicable durante	Efectos
Diseño de documento “fuente”	Preventivo	Captura de datos Preparación de datos	Reduce errores de grabación y acelera la captura de datos Reduce errores de tecleo y acelera la preparación de datos
Diseño de códigos de datos	Preventivo	Captura de datos Preparación de datos	Reduce errores de codificación y acelera la captura de datos Reduce errores de tecleo y acelera la preparación de datos
Dígitos de Control	De detección	Preparación de datos	Detecta errores de codificación y grabación
Controles “batch”	De detección	Captura de datos Preparación de datos	Detecta omisiones y /o modificaciones en la ejecución del batch Detecta altas, borrados o alteraciones de los datos
Controles de procedimientos	Preventivo	Captura de datos Preparación de datos	Reduce errores de grabación y acelera la captura de datos Reduce errores de tecleo y acelera la preparación de datos
Revisión de procedimientos	De detección	Captura de datos	Detecta errores de grabación de datos
Verificación	De detección	Preparación de datos	Detecta errores de tecleo
Diseño de entorno y tareas de tecleo	Preventivo	Preparación de datos	Reduce errores de tecleo y acelera la preparación de datos
Entrenamiento del personal	Preventivo	Captura de datos Preparación de datos	Reduce errores de grabación y acelera la captura de datos Reduce errores de tecleo y acelera la preparación de datos

Fig. 3.1 Controles de Captura y Preparación de Datos

3.1.2 Evaluación de los métodos de Preparación y Entrada de datos

Los métodos de preparación y entrada de datos han ido evolucionando con el tiempo. En un principio sólo se disponía de tarjetas perforadas, para las que el sistema de verificación se reducía a una doble digitación y al uso de dígitos de control en datos. Posteriormente se utilizaron las cintas de papel con los mismos controles que las tarjetas perforadas.

Más tarde aparecieron las cintas magnéticas, que eran dispositivos que permitían la captura de datos introducidos desde un teclado y posibilitaban controles de “paridad” de los datos, controles de lectura después de escribir, realización de copias de seguridad, relleno automático con ceros o espacios en campos numéricos y alfanuméricos, justificación de caracteres a la izquierda o a la derecha, etc.

Luego se sustituyeron las cintas magnéticas por los discos magnéticos, dispositivos para la recogida de datos con características similares a las cintas, aunque de mucha mayor velocidad, para llegar posteriormente a los cartuchos, cuya mayor ventaja frente a los anteriores era su facilidad manipulación.

Finalmente, se dispone de disquetes, CD Rom, dispositivos de reconocimiento de caracteres de tinta magnética, formados por matrices de puntos (MICR: Magnetic Ink Character Recognition) y utilizados por ejemplo en la numeración de cheques, lectores de caracteres ópticos (OCR: Optical Characters Readers), sensores de marcas ópticas (códigos de barras), terminales que pueden ser de teclado con salida a papel o con salida a pantalla (VDU: Visual Display Unit), TPV's (Terminales Punto de Venta) ya sean de código de barras o de precisión y, por supuesto, los ordenadores personales.

3.1.3 Diseño de documentos "fuente"

Los documentos fuente son formularios usados para captura de datos. Éstos van a ser muy importantes ya que un buen diseño puede reducir un número considerable de errores.

Para su creación es necesario contar con unas normas básicas de diseño:

- Preimprimir la información constante. Es decir, que aparezca preimpresa la información ya conocida, para evitar su digitación por parte del usuario.

- Proporcionar títulos, cabeceras, notas, etc. De esta forma se puede asociar la información y se aclara la tarea al usuario.

- Enfatizar y subrayar partes diferentes. Para captar la atención del usuario en los aspectos más importantes del documento fuente.

- Colocar bien los campos para facilitar su uso. Lo más normal es que los datos en el documento fuente aparezcan en el mismo orden en que van a ser luego introducidos en el ordenador a través de una pantalla de captura de datos. Así, se deben de poner los campos obligatorios en primer lugar. Además, se debe mantener una secuencia lógica a la hora de pedir los datos, por ejemplo, si se pide el nombre, pedir a continuación los apellidos y luego la dirección, y no primero el nombre, luego la dirección y por último los apellidos.

- Proporcionar respuestas múltiples, para elegir la más adecuada, siempre que sea posible.

- Utilizar cajas para identificar el tamaño de la respuesta. Esto ayudará al usuario a determinar lo que tiene que escribir. Por ejemplo en un campo tipo fecha, si se tiene una caja de 4 dígitos para el año, ya se sabe que el formato es en cuatro caracteres.

- Prenumerar los documentos. De esta forma es más fácil la identificación de los documentos que se están rellenando.

3.1.4 Controles de Codificación de Datos

En toda aplicación, va a ser necesario la utilización de códigos para la representación de los datos. Un código no es más que un identificador único, que permite localizar un item de datos más rápidamente que si lo identificáramos con texto.

Debido a su cualidad de identificador único, es necesario garantizar un buen diseño a la hora de crear un código, ya que un diseño ineficiente afectará al proceso de captura / entrada de datos de un modo significativo.

Por tanto, un buen código debe de ser:

1) Flexible: Que permita la posibilidad de añadir nuevas categorías y que sea adaptable a cambios.

2) Significativo. Que indique el valor de los atributos de la entidad a la que se refiere.

3) Compacto. Que sea lo más pequeño posible.

4) Conveniente. Que sea fácil de asignar, codificar, descodificar y capturar

No obstante, aunque un buen diseño ayuda a la captura de datos, no se puede garantizar que no se vayan a producir errores durante la misma. Los posibles errores son los siguientes:

- Adicción. Añadir un carácter extra. Ejemplo: Poner 879**1**42 en vez de 87942

- Truncamiento. Se omite un carácter. Ejemplo: Poner 8792 en vez de 879**4**2

- Transcripción. Se graba un carácter erróneo. Ejemplo: Poner 8**1**942 en vez de 87942

- Transposición. Se invierten caracteres adyacentes. Ejemplo: Poner **78**942 en vez de 87942

- Doble transposición. Se invierten caracteres separados por más de un carácter. Ejemplo: Poner 8**49**7**2** en vez de 87942

- Al azar. Alguna combinación de las anteriores. Ejemplo: Poner **A**7**49**2 en vez de 87942

Se han realizado investigaciones sobre los códigos y los errores cometidos y se ha llegado a la conclusión de que existen dos atributos determinantes del número de errores cometidos: la longitud del código y la mezcla de caracteres numéricos y alfabéticos.

En el primer caso, estudios de [Miller, 1956] concluyen que la memoria a corto plazo permite almacenar entre 5 y 9 caracteres (siete por término medio), y en el segundo caso se ha demostrado que se produce un menor número de errores si se agrupan los

caracteres numéricos por un lado y los alfabéticos por otro [Owsowitz & Sweetland, 1965]

3.1.5 Dígito de control

La digitación de un código erróneo (por ejemplo, equivocarse al introducir un número de cuenta) puede tener consecuencias graves. Por eso se deben de utilizar métodos de control que garanticen que esto no suceda. Uno de los métodos mas comunes es el uso de dígitos de control.

El método consiste en añadir un carácter redundante, ya sea como prefijo, sufijo o situado en cualquier otro sitio, a la cifra que se está capturando. Este dígito extra sirve para comprobar si los valores introducidos son válidos.

Existen muchos métodos de crear un dígito de control. El más fácil consiste en sumar los dígitos a capturar, pero tiene el fallo de no detectar la transposición. Por ejemplo, el dígito de control para el número 12345 sería 15, el mismo que el del número 54321.

Se verá a continuación un método bastante empleado para la creación de un dígito de control.

Sea el número 2148. Los pasos para crear el dígito de control son los siguientes:

- Multiplicar cada dígito por un peso (por ejemplo: 5-4-3-2)

  2 * 5 = 10
  1 * 4 = 4
  4 * 3 = 12
  8 * 2 = 16

- Sumar los resultados obtenidos: 42

- Dividir por un módulo (por ejemplo 11)

  42 / 11 = 3, y resto 9

- Restar el resto del módulo. El resultado será el dígito de control.

  11- 9 = 2

- Añadirlo como sufijo al número original: 2148**2**

Recálculo del dígito para detectar error:

- Multiplicar cada dígito por su peso. Al dígito de control se le asigna un peso de 1.

  2 * 5 = 10
  1 * 4 = 4
  4 * 3 = 12
  8 * 2 = 16
  2 * 1 = 2

- Sumar los resultados obtenidos: 44

- Dividir por el módulo

  44 / 11 = 4, y resto 0

- Si el resto es cero existe una alta probabilidad de que el número digitado sea correcto.

Puede suceder que el dígito de control obtenido tenga más de un carácter. En ese caso se puede tomar una de las decisiones siguientes:

1) Admitir dígitos de control de más de un carácter. No es muy aconsejable, porque aumenta la longitud del dato.

2) Convertir el dígito de control a un valor alfanumérico. Tampoco es muy aconsejable, ya que se produce una mezcla de caracteres numéricos y alfanuméricos.

3) Utilizar dos juegos de pesos, para que uno de ellos no produzca un dígito de control de más de una cifra. En este caso, también habrá que utilizar ambos juegos de pesos en el proceso de descodificación.

Por otra parte, si el número tiene valores alfabéticos se les puede asignar un valor numérico y ya se puede utilizar el método.

En general, el método del dígito de control tiene como inconvenientes el aumento de la longitud del dato, lo que puede llevar a problemas de almacenamiento y de tiempo de proceso, por lo que se debe usar sólo cuando sea necesario. También es preciso evitar su uso de un modo manual.

3.1.6 Controles sobre el Batch (Proceso por lotes)

Con estos controles buscamos dos propósitos: por un lado garantizar la precisión de su contenido y que éste está completo y por otro garantizar que no hay pérdidas durante la manipulación.

Cuando estamos ante procesos por lotes, se debe de tener en cuenta aspectos tales como: existencia de un número único para el lote, existencia de totales de control, fecha de preparación, información de errores detectados, sitio para firmas autorizadas, etc.

3.1.7 Resumen

Evaluar los controles de captura y preparación de datos es una labor compleja para el auditor ya que existe una gran combinación de métodos de captura, preparación y entrada de datos, cada una de ellas con sus ventajas e inconvenientes.

El auditor debe de conocer perfectamente los métodos para ser capaz de evaluarlos adecuadamente.

3.1.8 Ejercicios y casos

3.1.8.1 Calidad de la Información

Comentar la calidad de la siguiente información por pantalla, de un sistema de facturación:

#87AB649531G

Cobrar a	Enviar factura a
ACME, Inc. 13 Rue del Percebe La Coruña	ACME, Inc. 13 Rue del Percebe La Coruña

¿ES CORRECTA LA INFORMACION ANTERIOR,
Y SI LO ES, SE REQUIERE EL ENVIO EN < 10 DIAS?

3.1.8.2 Dígito de Control

Calcular el dígito de control con los siguientes datos:

Código	**Módulo**	**Peso**
753642	10	1-2-1-2-1-2
43196	11	6-5-4-3-2
841075	37	1-3-7-1-3-7

3.1.8.3 Uso del dígito de control

Los siguientes campos corresponden a un sistema de contabilidad general. Indicar en que campos se debería de utilizar un dígito de control. Razonar la respuesta.

Tipo de registro, número de cuenta, código de transacción, descripción, importe, número de documento "fuente".

3.2 Controles de acceso y comunicaciones

En cualquier aplicación va a ser necesario realizar dos tipos de controles de acceso: para evitar accesos o usos no autorizados y para mantener la integridad de los datos enviados o recibidos por una línea de comunicaciones. Todo ello es debido a la necesidad de proteger las transferencias electrónicas (entre bancos, vía Internet, etc.) y cualquier otro tipo de servicios on-line, como los de Banca Electrónica, por ejemplo.

3.2.1 Controles de Acceso

Los controles de acceso dependen en gran medida del medio en el cual se esté trabajando: si el ordenador es usado por una sola persona, entonces posiblemente no sean necesarios, pero si existen varios usuarios, entonces lo más probable será tener que implantarlos.

Los mecanismos utilizados son la identificación y la autentificación de los usuarios.

3.2.1.1 Identificación y autentificación

Para controlar el acceso a un sistema informático se debe de identificar a la persona que accede y comprobar que tiene permiso para hacerlo. Estos controles se realizan a través de la asignación de un código de usuario y de una palabra de paso o contraseña.

Sin ser exhaustivos, los conceptos relacionados con estos controles son los siguientes:

1) Privilegios de acceso. Son aquellos permisos de los que dispone un usuario determinado o un grupo de usuarios.

2) Tipos de permisos. De creación, lectura, modificación y de borrado. Es decir, las acciones que puede hacer un usuario determinado.

3) Autorizaciones. Pueden ser a toda una Base de Datos, a parte de una Base de Datos, a un registro individual, o incluso a ciertos campos de un registro.

4) Privilegios en cascada. Un usuario que tiene un cierto privilegio puede otorgarlo (si tiene permiso para ello) a otros usuarios. Si pierde su privilegio lo pierden todos los usuarios que lo habían obtenido de él.

5) Contraseñas. Deben de guardarse encriptadas. Deben de tener una longitud mínima (6 ú 8 caracteres) No deben de estar formadas por palabras que tengan significado. Deben de combinar letras y números, así como por mayúsculas y minúsculas. No se deben de poder repetir durante un cierto período de tiempo y debe de ser obligatorio cambiarlas cada cierto tiempo. Es necesario implantar un sistema para rechazar una conexión tras un número determinado de intentos fallidos (normalmente tres) al introducir la contraseña. Un buen sistema para crear una contraseña podría ser elegir las dos primeras letras de cada palabra de una frase que se conozca.

3.2.2 Controles de Comunicaciones

Si se utiliza una Red para las comunicaciones, el auditor debe evaluar las capacidades de ésta para ver si cumplen los objetivos de AI.

Alguno de los aspectos relacionados con las comunicaciones que el auditor debe de conocer son los siguientes:

1. Protocolos de detección de errores.

- Chequeo de bucle
- Redundancia (paridad, códigos M de N, códigos cíclicos, etc.)

2. Protocolos de corrección de errores.

- Códigos de corrección de errores o mensajes redundantes
- Retransmisión de datos erróneos (ACK-NAK, etc.)

3. Hardware y Software de Red

Además, para garantizar la rapidez de las comunicaciones va a ser importante la elección de hardware y software adecuados, y conocer aspectos tales como la topología de la Red (estrella, anillo, bus, etc.), los protocolo de comunicaciones (CSMA/CA, CSMA/CD, Token-passing, etc.), y los dispositivos hardware utilizados, tales como modems, routers, maus, hubs, etc., etc.

Como se puede apreciar, la labor de AI es amplia y compleja, y el auditor informático debe de tener unos grandes conocimientos de informática para poder desarrollar su trabajo.

3.2.3 Criptografía

Es una parte de la Criptología (ciencia de los códigos secretos), que estudia la transformación de datos en códigos sin significado para quien no posea la clave para descifrarlos.

3.2.3.1 Técnicas criptográficas

La fuerza de una técnica criptográfica se mide en términos del Factor de Trabajo, es decir, el tiempo y coste necesario para descifrar el texto encriptado. Una técnica criptográfica transforma (encripta) datos, conocidos como "texto plano", en criptogramas (texto cifrado o encriptado)

Hay tres clases de técnicas criptográficas:

- **Transposición.** Es la técnica más sencilla, por lo que no es conveniente utilizarla, ya que es relativamente fácil de descifrar. Consiste en cambiar el orden de los caracteres de acuerdo con una regla predeterminada. Por ejemplo, cambiar la posición de los caracteres en pares consecutivos (el carácter b representa un espacio en blanco)

  Así: LA PAZ ES NUESTRO OBJETIVO
  Será: AL Pb ZA Eb bS UN SE RT bO BO E J IT OV

- **Substitución**. Reemplazar los caracteres con otros de acuerdo con un patrón dado. El ejemplo más típico es el "alfabeto Cesar"

  - Se elige una clave
  - Se substituyen las primeras letras del alfabeto por las de la clave
  - El resto se substituyen por las letras del alfabeto no utilizadas y consecutivas.

  Texto plano: ABCDEFGHIJKLMNÑOPQRSTUVWXYZ
  Clave: ESMUYFACIL
  Resultado: ESMUYFACILBDGHJKNÑOPQRSTVWX

  Así: LA PAZ ES NUESTRO OBJETIVO
  Será: DE NEX YO HRYPQOK KSLYQISK

  Es también un algoritmo muy sencillo, por lo que no se debe de utilizar si la seguridad es primordial.

- **Producto**. Es una combinación de las dos técnicas anteriores. Es más resistente al descifrado, y el método más importante utilizado en la actualidad. Los puntos que vienen a continuación se refieren exclusivamente a este tipo de cifrado.

  1 Sistema de cifrado

  Todo sistema de cifrado va a estar compuesto por un método de cifrado (algoritmo de cifrado), que es la técnica criptográfica básica, y una clave criptográfica sobre la que opera el algoritmo junto con el texto plano para obtener el texto cifrado.

  2 Características deseables

  Cualquier sistema de cifrado debe de cumplir unos requisitos básicos. Estos son:

- Factor de trabajo alto. Es decir, que descifrar la clave requiera mucho esfuerzo, ya sea de tiempo o de recursos.

- Clave corta. Para que se pueda cambiar con frecuencia y fácilmente.

- Simplicidad. Para evitar costes y facilitar los cambios.

- Baja propagación del error. Algunos tipos de texto cifrado dependen de otro texto cifrado previo, generado por un mensaje. Si se usa un método de encriptado en cadena, la corrupción de un solo bit del texto cifrado causará un error en el subsiguiente desencriptamiento.

- Baja expansión del tamaño del mensaje. Algunos sistemas de cifrado introducen ruido en el mensaje para evitar que el uso de técnicas estadísticas pueda romper el código. Esas técnicas examinan las frecuencias de repetición de una letras, de pares de letras, etc.

Estas características no se pueden conseguir simultáneamente cuando se encripta lenguaje natural. Por lo tanto, se debe llegar a un compromiso entre el tamaño de la clave y simplicidad del algoritmo, por ejemplo. Así, se tienen dos tipos de sistemas diferentes: los sistemas que usan un algoritmo simple y clave larga, denominados sistemas de clave larga y los que usan un algoritmo conocido por su fuerza, denominados sistemas de algoritmo fuerte**.**

El Algoritmo DES

En 1997, el National Boureau of Standards (NBS) adoptó como estándar el algoritmo DES (Data Encryption Standard), desarrollado por IBM, del que existen implementaciones hardware y software. [Bright & Enison, 1976] [Lennon, 1978]

Las características principales de este algoritmo son las siguientes:

1) Es un algoritmo público.

2) Sistema de cifrado de algoritmo fuerte. Los sistemas de clave larga no se pueden romper si la clave es aleatoria e igual en longitud al número de caracteres del mensaje a encriptar. No obstante, en la mayoría de los Sistemas de Proceso de Datos no se pueden emplear estas claves por el gran volumen de tráfico que originan, por lo que éstas tienen que ser relativamente cortas, de longitud fija y permitir uso repetitivo. Por estas razones la NBS eligió como estándar un sistema de algoritmo fuerte.

3) Clave de 64 bits (56 + 8 de paridad)

4) Convierte un bloque de 64 bits (8 caracteres) en un bloque de 64 bits de texto cifrado tras 16 pasadas de cifrado.

3.2.3.3 Funciones del auditor.

La tarea del auditor es garantizar que el sistema de cifrado sea seguro. Para ello debe de verificar que la clave se mantenga en secreto, ya que el algoritmo puede no ser seguro a lo largo del tiempo o bien ser un algoritmo público, como en el caso del algoritmo DES.

Para garantizar que las claves se mantengan en secreto, se debe implantar un sistema de gestión de claves (SGC), y garantizar aspectos como la generación, la distribución y la instalación de las claves. Otra función del auditor es la evaluación periódica del SGC, ya que esta evaluación constituye el aspecto más crítico para comprobar la fiabilidad de un algoritmo de encriptación.

3.2.3.4 Sistema de Gestión de Claves (SGC)

En todo SGC, existen una serie de aspectos a tener en cuenta por el auditor:

1. Generación de claves

- Si la clave es única, existe una posibilidad alta de conocerla o descifrarla (por ejemplo analizando las salidas, conocidas las entradas) por personal del CPD, lo que implicaría el descifrado del texto, por lo que es muy conveniente usar claves múltiples. [Ehrsam et al, 1978]

- Las claves se deben de generar las claves completamente al azar. No sirve utilizar generadores de números pseudo-aleatorios.

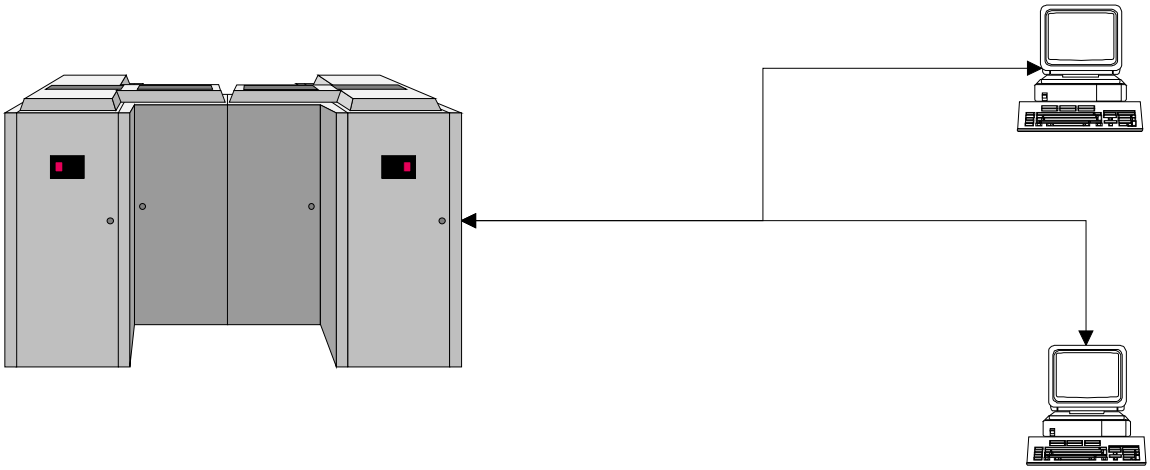
Existen dos tipos de claves:

- Claves para encriptar claves. Que pueden permanecer relativamente estables.

- Claves para encriptar datos. Que deben de ser variables en el tiempo y se deben de cambiar dinámicamente. Estas claves existen sólo mientras existen los datos.

En el siguiente diagrama se muestra un ejemplo típico.

El único punto débil de este sistema es, obviamente, la HMK. Esta debilidad se puede minimizar con el uso de múltiples HMK.



Host Master Key (HMK) + Algoritmo	Sesion Key (SK) (para datos)	Terminal Master Key (TMK)
Deben de ser inaccesibles, excepto Para encriptar y desencriptar. Hay una copia de la TMK encriptada Con la HMK.	La SK se genera automáticamente al comenzar cada sesión. El Host desencripta su TMK y encripta la SK con la TMK y la envía al terminal, donde es desencriptada con la TMK del terminal.	

2. Distribución de claves

La distribución de las claves va a depender del tipo de clave. Si están encriptadas, se pueden transmitir por una línea de transmisión de datos. En caso contrario, se pueden enviar por correo certificado, por teléfono enviándolas fragmentadas, etc. Va a ser vital mantenerlas en secreto, para mantener la seguridad del sistema.

3. Instalación de las claves

Si las claves no se generan internamente, es necesario instalarlas. Además, en el caso de claves TMK, hay que asegurarse de que ambas copias sean idénticas.

El método de instalación dependerá del hardware y software disponible.

3.2.3.5 Criptografía para Bases de Datos

Cuando se trabaja en un entorno de Bases de Datos, es necesario hacer algunas consideraciones especiales. En primer lugar, no valen todos los tipos de encriptación, ya que, por ejemplo, la encriptación en cadena crea interdependencias de bits (Encriptación en cadena: clave para encriptar clave, que encripta a su vez a otra clave, etc.)

Además, se debe auditar los procesos de altas, bajas, modificaciones y consultas, ya que pueden hacer irrecuperable el texto plano. Por este motivo, y por cuestiones de rendimiento (tiempos de respuesta), sólo se deben de encriptar los datos críticos.

Por otra parte, es necesario controlar los cambios en la HMK, ya que éstos van a afectar a todas las claves encriptadas con ella. Habrá que desencriptar todas estas claves y volverlas a encriptar con la nueva HMK, proceso normalmente largo y consumidor de recursos. [Ehram et al, 1978]

3.2.4 Resumen

Es necesario implantar controles de acceso para evitar usos no autorizados a los sistemas de proceso de datos. Estos controles se realizan a través de la identificación y autentificación de los usuarios.

También es necesario disponer de, así como de controles de comunicaciones para preservar la integridad de los datos durante su transmisión por las líneas de comunicación de datos.

El auditor debe conocer las redes de comunicaciones de los datos, y estar familiarizado con aspectos tales como las técnicas de Criptografía empleadas, los algoritmos de encriptación de datos, la generación y tratamiento de las claves, etc.

3.2.5 Ejercicios y Casos

3.2.5.1 Pago telefónico

Eres el gerente de Auditoría Interna de un banco que ha decidido instalar un sistema de pago por teléfono para sus clientes. El sistema permitirá a los clientes llamar a un número determinado para acceder, grabar las transacciones de pago que desean hacer dentro de los próximos 30 días y hacer transferencias entre distintas cuentas. Un sistema asistido por voz grabada indicará a los clientes cómo realizar las diversas operaciones, y todos los datos se darán de alta por medio de un teléfono del tipo touch-tone.

El gerente de Proceso de Datos te ha pedido tu consejo sobre los controles de acceso que consideras que debería de tener el nuevo sistema.

Se pide:

Preparar un breve informe con tus recomendaciones, teniendo en cuenta, entre otros, los siguientes aspectos:

1. Saber si se está tratando con un cliente válido.
2. Saber si el cliente solo transfiere fondos desde o hacia cuentas para las que está autorizado.
3. Comprobar que el cliente no sobrepasa sus saldos.
4. Verificar que los pagos se hacen solo a deudores previamente autorizados por el cliente.

3.2.5.2 Reserva de billetes

Global Airways es una línea aérea con base en Los Angeles que dispone de un sistema dedicado a reservas de billetes, con mas de 10.000 terminales repartidos por todos los

Estados Unidos, conectados a un ordenador central (Algo así como Iberia, pero a lo bestia)

Eres un miembro del equipo de Auditoría Externa que está examinando los controles de comunicaciones y de acceso al sistema, y estás asombrado de que éste no posea un mecanismo de protección de acceso por palabra de paso.

Cuando le preguntas al gerente de Proceso de Datos por esta peculiaridad, éste te cuenta que las palabras de paso no son necesarias para este sistema. Te explica que cada terminal conectado al ordenador central está provisto de un número de identificación único, el cual está almacenado en una tabla en un área segura del sistema operativo. Este número debe de ser suministrado por el terminal y acompañar a cada mensaje que envía, ya que de lo contrario el mensaje sería rechazado por el sistema. Además, el número lo envía el terminal automáticamente, ya que se encuentra cableado en el propio terminal.

Además, el gerente te cuenta que ya se ha utilizado anteriormente un sistema basado en palabras de paso, que se tuvo que abandonar por ineficiente. A cada administrativo que manejaba un terminal se le había asignado una palabra de paso, pero como un mismo terminal era utilizado por varios administrativos, el sistema era muy incómodo de manejar, ya que cada vez que había un cambio de personal era necesario realizar una desconexión y una conexión, por lo que el sistema tenía problemas durante las horas punta.

Se pide:

Escribir un informe identificando cualquier cosa que pueda ir mal en el sistema (si es que hay algo que pueda ir mal), causando pérdidas de bienes o violación de la integridad de los datos.

3.3 Controles de Entrada

El objetivo de estos controles es detectar errores en los datos que entran a un Sistema Informático y se deben aplicar en la preparación de los datos (si los dispositivos lo permiten) y en la entrada de datos.

En general las validaciones de los datos se deben de hacer lo antes posible.

3.3.1 Controles de Validación de Entradas

Se realizan a cuatro niveles: de campo, de registro, de procesos batch y de fichero.

3.3.1.1 Controles de Campo

En los campos de un fichero se deben de controlar los datos omitidos, la presencia de espacios en blanco, el tipo del campo (alfabético, numérico, etc.), los rangos de los datos, la pertenencia a un conjunto de datos, el dígito de control, etc.

3.3.1.2 Controles de Registros

Los controles a realizar en los registros de un fichero son los siguientes:

1. Datos razonables. El valor de un determinado dato debe de estar de acuerdo con las normas especificadas para ese dato. Por ejemplo, un cierto campo puede pasar un control de rango, pero otro dato del mismo registro puede indicar que el valor de ese campo no es razonable. Así, el campo “sueldo” puede admitir un valor de 18 millones, por lo que pasaría el control de rango válido, pero otro campo de ese registro indica que se trata de un administrativo, por lo que el sueldo sería inválido, ya que un administrativo no tiene un sueldo de 18 millones. Es necesario establecer controles cruzados de este tipo.

2. Signo válido. Por ejemplo, una retirada de dinero en una cuenta debe de tener signo negativo.

3. Tamaño. Es necesario comprobar la longitud de los registros, cuando éstos son de longitud variable.

4. Secuencia. Cuando un registro lógico tiene más de un registro físico, se debe comprobar la secuencia de aparición de los distintos tipos de registro. Por ejemplo, si se tiene un registro tipo 1 con datos personales, otro tipo 2 con datos financieros y un último registro tipo 3 con datos técnicos, es necesario comprobar que, por cada registro lógico de la persona, existen todos los registros físicos.

3.3.1.3 Controles de procesos Batch

En este tipo de procesos por lotes, deben de existir los siguientes controles:

1. Totales: ¿Cuadra la suma de los lotes parciales con el registro de totales?

2. Secuencia: Si existe un número de secuencia de registros, ¿éstos están en ese orden?

3. Tamaño: Si existe un número máximo permitido de registros, ¿éste se sobrepasa?

3.3.1.4 Controles de Fichero

A nivel de fichero se deben de controlar aspectos como el label interno, el número de generación del fichero, la fecha de retención (fecha a partir de la cual el fichero deja de ser válido) y los totales de control, entre otros.

3.3.2 Diseño del programa de entrada de datos

Un programa de entrada bien diseñado, garantiza la calidad de los datos; si se hace validación intensiva, hay que tener un buen diseño para que el programa sea eficiente; si se procesan grandes volúmenes de datos, se considerarán aspectos de eficiencia.

El auditor se debe centrar en cómo se validan los datos, cómo se tratan los errores y cómo se informa de los mismos.

3.3.2.1 Validación de los datos

Para realizar una correcta validación de los datos de entrada es necesario:

1. Determinar en primer lugar lo que es correcto y luego las posibles desviaciones
2. Determinar una lógica de validación agrupada en módulos (Programación Estructurada)
3. Ordenar los errores por probabilidad de aparición
4. Validar todos los errores posibles (idealmente)
5. Comprobar los valores y códigos que estarán en tablas y que no deberán de estar integrados en el código
6. Identificar todos lo errores posibles en una sola pasada
7. Validar exhaustivamente las condiciones
8. Realizar una corrección automática de errores si es posible
9. Comprobar la documentación del programa

3.3.2.2 Tratamiento de los errores

Cuando se producen errores no es suficiente realizar un informe de los mismos sino que es conveniente almacenarlos en un fichero de errores, que se debe de actualizar y mantener. Una vez corregidos los errores en el fichero de errores, éste se utiliza como fichero de entrada de datos a la aplicación.

3.3.2.3 Informe de errores

El informe de errores debe de identificar claramente los errores, los cuales se deberán de ordenar antes de imprimir el informe. De esta manera, al tenerlos ordenados por tipo de error, es más fácil asignarlos a las distintas personas encargadas de corregirlos, y aunque se trate de una única persona, el tenerlos clasificados le facilitará el trabajo.

Además se tendrá que documentar todos los errores de un mismo registro e indicar la causa del error, si es posible, para posteriormente obtener una estadística de los errores encontrados, por tipo de error, con la frecuencia de cada error, totales de control, etc.

3.3.3 Control sobre la entrada de datos

El objetivo del control es garantizar que:

- Se introducen todos los datos: Para comprobarlo, utilizar registros y totales de control e imprimir un informe con todos los datos procesados.

- Se corrigen todos lo errores: Verificar que los errores corregidos aparecen en los informes de errores.

- Los errores no se corrigen más de uno a la vez: Verificar que los errores corregidos aparecen en los informes de errores.

- Se identifican los cambios en los patrones de los errores: Obtener estadísticas con la frecuencia de los diferentes errores.

- Existen copias de seguridad de los datos de entrada: Verificar la existencia de procedimientos de copias de seguridad y recuperación.

3.3.4 Resumen

En primer lugar, se ha determinado la necesidad de establecer controles de entrada a cuatro niveles: de campo, de registro, de los procesos batch y de fichero.

A continuación se ha estudiado el diseño del programa de entrada de datos, encargado de determinar cómo se validan los datos, cómo se tratan los errores y cómo se informa de los errores.

Y para finalizar, se han revisado los controles sobre la entrada de datos y sus características principales.

3.3.5 Ejercicios y casos

3.3.5.1 Sistema de pedidos

Tu labor es la de auditor interno colaborando en la fase de diseño de un sistema nuevo de pedidos. El programador responsable del diseño de las validaciones de los datos de entrada te pide tu opinión sobre si los tests de validación de entradas propuestos son adecuados. Los controles propuestos son los siguientes:

	Datos omitidos	Debe ser numérico	Debe ser alfabético	Rango válido	Dígito de control	Código válido	Signo válido
Número de cliente		X		X	X		
Número de vendedor		X				X	
Número de pedido	X						
Número de pieza	X				X		
Cantidad pedida	X	X				X	
Instrucciones de precio			X			X	

Los siguientes datos son relevantes para la toma de decisiones:

Campo	**Descripción**
Número de cliente	Valor numérico entre 01000 y 90000.
Número de vendedor	Debe de ser uno de los 50 valores numéricos existentes.
Número de pedido	Cinco caracteres, el primero alfabético y los restantes numéricos.
Número de pieza	Campo alfanumérico.
Cantidad pedida	Campo numérico de cuatro caracteres.
Instrucciones de precio	Alfabético; solo son válidos cuatro códigos.

Se pide:

Escribir un breve informe con tus comentarios sobre los tests de validación.

3.3.5.2 Control de personal

El registro siguiente proporciona información sobre tiempos de empleados para la confección de la nómina de una empresa:

Campo	**Formato y tamaño del campo**
Número de empleado	Numérico, 6 caracteres.
Horas normales	Numérico, 2 caracteres.
Horas extras	Numérico, 2 caracteres.
Gastos / comisiones	Numérico, 6 caracteres. Cuatro enteros y dos decimales.
Ausencias por enfermedad	Numérico, 2 caracteres.
Vacaciones	Numérico, 2 caracteres.

Se pide:

¿Qué controles de validación se deben de establecer sobre cada campo? Suponer los valores de los datos que sean necesarios para realizar las pruebas, siempre que sean valores razonables. Todos los campos son de longitud fija.

3.4 Controles de Proceso

El Objetivo es detectar errores, lógicos o de hardware, durante el periodo de tiempo comprendido entre la lectura de los datos y la salida de los resultados.

3.4.1 Controles de Validación

Se realizarán fundamentalmente sobre campos numéricos, ya que el resto de campos normalmente requieren pocos controles.

Los más comunes se indican en la tabla siguiente.

Nivel de control	**Tipo de control**	**Explicación**
Campo	Desbordamiento (overflow)	Falta inicializar campos, tablas, etc.
Registro	Rango; controles cruzados	Rango aplicable a un campo; el valor de un campo puede condicionar el de otro.
	Signo	El contenido de un campo puede determinar el signo de otro.
Fichero	Acarreo de totales	Se pueden calcular de modo independiente, y comprobarlos al final del batch. Por ej.: Sueldo = bruto + extras – deducciones
	Totales	Ej.: Saldo = Entradas – Salidas

Fig. 3.2 Algunos controles de validación durante el proceso

3.4.2 El "estilo" de Programación

El uso de Programación Estructurada facilita notablemente la labor del auditor para la detección de errores de programación. No obstante, el auditor debe de saber que existen ciertas causas de error muy conocidas, que se deben validar, por lo que deberá de preparar una lista exhaustiva de ellas. Como muestra, una lista de chequeo podría contener las siguientes comprobaciones:

1. En actualizaciones secuenciales, comprobar que el fichero de transacciones se ha clasificado adecuadamente antes de emparejarlo con el fichero maestro.

2. Asimismo en actualizaciones secuenciales, comprobar que se procesan todos los registros, ya que el último registro se suele "perder" si el código no es correcto.

3. Verificar los redondeos. Es muy fácil que se produzcan discrepancias si se emplean distintos tipos de redondeo en totales que se deban de comparar.

4. Etc.

3.4.3 Control de Concurrencia.

Se deben tener en cuenta aspectos como el abrazo mortal, sus soluciones y los métodos de prevención. En la tabla siguiente se muestran algunas estrategias y su explicación.

Estrategia	**Explicación**
Presecuenciar procesos	Si se sabe qué procesos producen el abrazo mortal (deadlock), ejecutarlos secuencialmente para evitar concurrencia.
Liberar recursos	Un proceso puede forzar a otro para que libere los recursos que tiene asegurados
Preordenar recursos	Para evitar una cadena circular de peticiones
Preseleccionar recursos	Un proceso puede obtener un control “exclusivo” de sus recursos antes de utilizarlos.

Fig. 3.3 Estrategias de control de concurrencia

3.4.4 Integridad del software de sistemas.

AI deberá de garantizar la integridad del software de sistemas y eso no es precisamente una tarea fácil. Cada vez se dan más casos de violación de la integridad del software del sistema. Las modificaciones de los proveedores del software pueden llegar a ser “problemáticas” y, además, existen herramientas especiales para programadores de sistemas que se pueden utilizar fraudulentamente.

Por tanto, es necesario incluir una revisión periódica del software de sistemas en el plan general de AI.

3.4.4.1 Integridad del Sistema Operativo.

El SO es el software más crítico de toda la instalación. Si se consigue acceder al SO se puede controlar el resto de los sistemas. Se podría, por ejemplo, asignar un fichero a un usuario no autorizado, conceder privilegios de acceso del mas alto nivel a un determinado usuario o grupo de usuarios, permitir la ejecución de programas no autorizados, etc.

Por otra parte, es muy difícil para el auditor encontrar la documentación necesaria para llevar a cabo esta auditoría, motivo por el cual será muy complicado adquirir los conocimientos necesarios para llevarla a cabo. Esto es debido, en parte por la dificultad inherente a los sistemas operativos, y en parte a que, frecuentemente, no se publica información sobre seguridad del sistema, precisamente para evitar que ésta se pueda ver en peligro.

3.4.4.2 Amenazas del SO.

Las amenazas que puede sufrir el SO son de dos tipos: accidentales, motivadas por fallos de hardware o de software, y deliberadas, es decir, accesos no autorizados. Dentro de este segundo tipo de amenazas podemos destacar:

- **Browsing** (Búsquedas de información) En residuos, basura, etc. para encontrar material (cintas, disquetes), que proporcionen información para acceso.

- **Masquerading** (Enmascaramiento) Acceso no autorizado suplantando a un usuario autorizado o al propio sistema. En este caso, se enviará un mensaje al operador simulando ser un mensaje del sistema, para que el operador haga algo.

- **Piggybacking** (Interceptación de comunicaciones) Interceptar mensajes entre el operador y el sistema y modificarlos o sustituirlos por otros, desde un terminal conectado a la línea.

- **Between lines entry** (Acceso entre líneas) Se usa el tiempo durante el cual un usuario autorizado está conectado e inactivo. Se necesita un terminal conectado a la línea.

- **Spooling** (Simulación del SO) Se hace creer al usuario que está interactuando con el SO, cuando en realidad lo está haciendo con un programa no autorizado. Un ejemplo típico consiste crear un programa que imite el procedimiento de conexión de un usuario con el sistema. De esta manera se captura la clave del usuario y su palabra de paso, para posteriormente simular una caída del sistema para forzar una nueva conexión del usuario, esta vez con el SO.

- **Trojan horse** (Caballo de Troya) Existen varias modalidades. Una podría consistir en modificar una utilidad del SO de manera que, cuando se use, se le asigne a un usuario falso el permiso de acceso más alto, por ejemplo.

3.4.4.3 Fallos del SO.

Evidentemente existen y pueden ser usados para violar la integridad de los sistemas. Entre otros, se tienen los siguientes::

- **Validación incompleta de parámetros.** El sistema no comprueba la validez de todos los atributos de una petición de usuario. Por ejemplo: se solicita una dirección de memoria fuera del área asignada a programas de usuario y el SO la concede.

- **Validación inconsistente de parámetros.** Cuando se aplican distintos criterios para validar el mismo constructor dentro del sistema. Por ejemplo: se puede crear una palabra de paso en blanco y luego no se puede borrar porque el SO la considera inválida.

- **Validación asíncrona.** Es una característica de la cual es posible aprovecharse para violar la integridad del sistema. Por ejemplo, un usuario emite una petición de entrada/salida y el SO valida la petición. A continuación el SO intenta asignar un canal pero no encuentra ninguno libre y emite una interrupción; el usuario aprovecha este momento para cambiar la dirección de entrada/salida, forzando al SO a devolver el control a una dirección ilegal.

- **Control de acceso inadecuado.** Verificación incompleta del SO. Por ejemplo, un usuario carga un programa que se llama igual que una rutina del sistema y éste no

verifica que se trate de una rutina propia que se haya cargado desde las librerías del sistema, y permite su ejecución.

- **Errores lógicos.** Se descubre un "bug" que se puede utilizar para dar permisos privilegiados a un determinado usuario, por ejemplo.

3.4.4.4 Requisitos de un SO. seguro

El SO es la parte más importante de un sistema de proceso de datos, y por lo tanto debe estar protegido, y no solamente frente a ataques externos. A continuación se indican las características que debería de tener un Sistema Operativo "seguro". [Stepczyk, 1974]

1. El SO debe de estar protegido frente a procesos de usuario. Un proceso de usuario no puede parar el sistema, destruir información esencial, tomar control del sistema, etc.

2. Los usuarios deben estar protegidos de otros usuarios. Ningún usuario puede estropear datos o programas de otro usuario.

3. Los usuarios deben de estar protegidos de si mismos. Un módulo de un proceso de usuario no puede corromper a otro módulo.

4. El SO debe estar protegido de si mismo. Un módulo de un proceso del SO no puede corromper a otro módulo del SO.

5. El SO debe de estar protegido del entorno, frente a incendios, inundaciones, caídas de tensión, etc.

3.4.5 Control sobre el mal funcionamiento del hardware.

Los problemas del mal funcionamiento del hardware, normalmente no son controlados por el auditor, a menos que se trate de equipos muy viejos. La propia tecnología ya proporciona gran fiabilidad para evitar errores. (Elementos de hardware redundante, grabación de datos en varios dispositivos a la vez, verificación automática del hardware, etc.)

3.4.6 Controles de Relanzamiento y puntos de verificación.

El auditor debe comprobar la existencia de estos controles y que los programas los incorporan. Por ejemplo la existencia de imágenes "after" y "before" en entornos transaccionales, existencia de Logs, etc.

3.4.7 Resumen.

Los controles de proceso intentan evitar los problemas ocasionadas por fallos de lógica de programación o errores del hardware o del software del sistema.

Detectar y corregir problemas del SO es difícil porque normalmente no hay demasiada documentación, por ser esta confidencial.

Hoy en día, los fallos de hardware son raros y el auditor no suele preocuparse por controlar el hardware.

El uso de Programación Estructurada y de procedimientos de Restart / Checkpoint ayudan en gran medida al auditor.

3.4.8 Ejercicios y casos

3.4.8.1 Totales incorrectos

Los totales de control de pesetas del fichero de transacciones de entrada y del fichero maestro de entrada, de una actualización secuencial de un fichero maestro, son correctos. No obstante, son incorrectos los del fichero maestro de salida.

Se pide:

Escribir un informe indicando una lista de las posibles causas de los totales incorrectos. Ordenar las causas por probabilidad de originar el fallo, indicando las más probables primero. Asimismo indicar cómo se podrían comprobar las posibles causas del fallo mostradas en la lista.

3.4.8.2 Particionamiento de la Base de Datos

Pieces & Parts, Ltd. es una empresa de manufacturas con sede en Nueva York, con fábricas en todo el país. La empresa piensa cambiar sus operaciones centralizadas de proceso de datos por operaciones distribuidas, pensando que las fábricas operarán de una forma más eficiente si cada una tiene sus propias facilidades de proceso de datos.

La mayor preocupación de los directivos es saber si la Base de Datos de la empresa se debería de particionar y ubicar las diferentes particiones en las fábricas que tengan una mayor probabilidad de necesitarlas, o bien si se deben de realizar periódicamente copias completas de la Base de Datos y enviarlas a las distintas fábricas.

Se pide:

Como jefe del departamento de Auditoría Interna, la gerencia te ha solicitado un informe pidiendo que se identifiquen las ventajas e inconvenientes de ambas posibilidades, desde el punto de vista de garantizar la precisión y el procesamiento completo de los datos.

3.5 Controles de Salida

Los objetivos de los controles de salida son preservar la integridad de los datos de los informes (independientemente del medio en que estén almacenados) y garantizar un uso efectivo de los informes.

En general, un sistema batch necesita un mayor control que un sistema on-line. En un sistema batch, un informe debe de recorrer un largo camino, desde que se genera hasta que se distribuye, y todo este recorrido debe de estar controlado por el auditor.

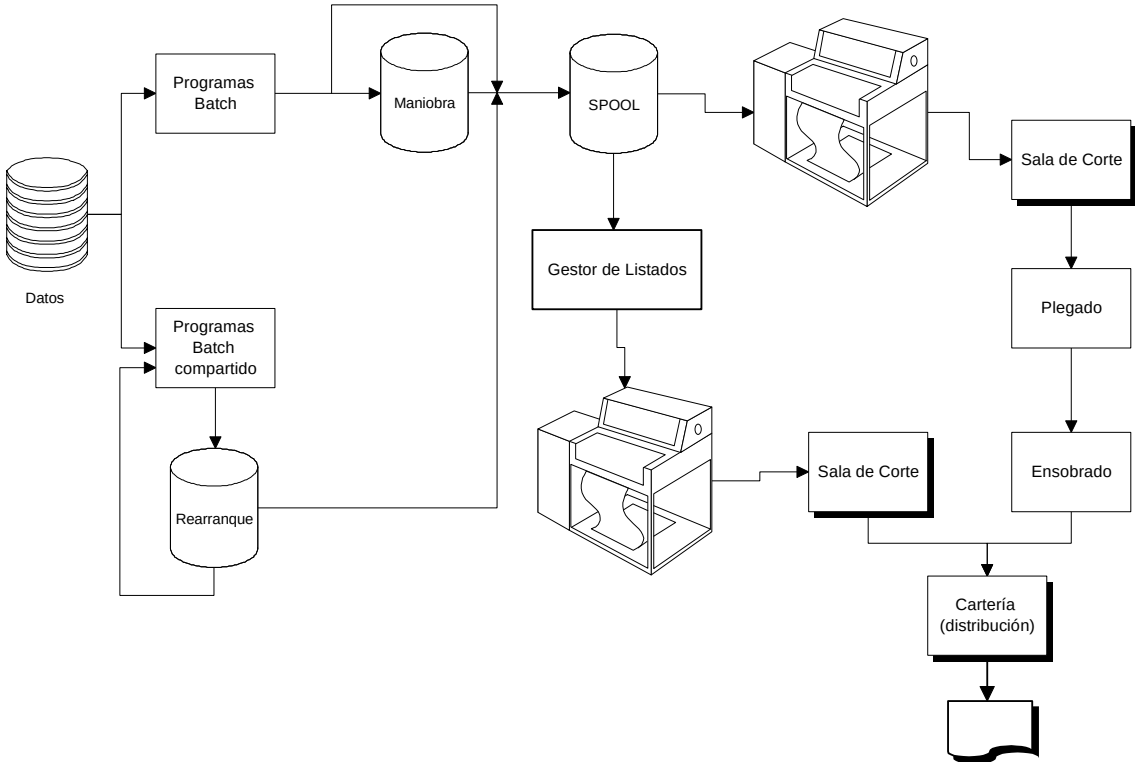


Fig. 3.4 Sistema de Gestión y Distribución de Informes

3.5.1 Controlando la salida del batch

Se debe cuidar el diseño de los informes y la gestión (creación, distribución y uso) de los informes. En la tabla siguiente se muestra la información de control que debe de tener un informe.

Información de control	**Lugar en el informe**	**Propósito**
Nombre	Página de título (PT)	Identificar el informe
Fecha y hora	PT y Página de detalle (PD)	Evitar confusión si se obtiene más de una vez
Lista de distribución (incluye el número de copias)	PT	Facilita la distribución
Período de proceso cubierto	PT	El usuario puede saber qué datos se han incluido en el informe
Programa que obtiene el informe	PT	Identificar el programa para modificaciones
Clasificación de seguridad	PT	Avisa a los operadores de la confidencialidad de los datos
Fecha de retención	PT	Fecha antes de la que el informe no debe ser destruido
Método de destrucción	PT	Si se necesitan procedimientos especiales para eliminar el informe
Cabecera de página	PD	Contenido de las columnas del informe
Número de página	PD	Evita que se quiten hojas del informe
Marca de fin de informe	Inmediatamente después de la última línea. Última página del informe.	Evita que se quite la última página del informe y que se añada información en ésta página.

Fig. 3.5 Información de control de un informe.

3.5.1.1 Control sobre formularios

AI debe de realizar controles sobre los formularios en blanco o preimpresos, sobre todo en éstos últimos, ya que normalmente contienen información de la empresa (NIF, dirección, etc.)

Entre otros, se tendrá un sistema de control de acceso al lugar donde están almacenados y un control periódico de inventario.

3.5.1.2 Control sobre programas de ejecución

AI debe de garantizar que cuando se ejecuta un programa, sea la versión correcta la que se ejecuta, y que no existen alteraciones vía consola. Además, comprobará la existencia de puntos de relanzamiento y recuperación.

3.5.1.3 Control sobre ficheros de impresión (SPOOL)

AI debe de controlar posibles modificaciones no autorizadas del SPOOL, copias no autorizadas, que cada fichero se imprima una sola vez, y que si se guardan copias su seguridad esté garantizada.

3.5.1.4 Control sobre la impresión

AI controlará aspectos tales como que sólo se realice el número de copias autorizado y que personal no autorizado pueda ver los datos confidenciales.

3.5.1.5 Control sobre la acumulación de informes

AI debe de controlar que no permanezcan en la sala de impresión durante mucho tiempo. Asimismo, se deberá de disponer de una lista de los informes que se obtienen durante cada turno de operadores.

3.5.1.6 Controles de Revisión de Informes

Se establecen con el fin de detectar errores obvios (datos no razonables, errores de formato, datos omitidos, etc.)

Estas revisiones se deben hacer periódicamente y al azar, y deben de comprobar a fondo las salidas. Si los datos son confidenciales, se debe comprobar además los permisos de las personas autorizadas para leerlos.

3.5.1.7 Controles de distribución de informes

Intentan garantizar que los informes sólo lleguen a los destinatarios autorizados, entregándolos en mano si es necesario. Asimismo se deberá de comprobar que no se realicen copias ilegales de los informes. Estos controles de entrega se deben de revisar periódicamente.

3.5.1.8 Controles de retención / almacenamiento

Se debe garantizar que los documentos se destruyen cuando ya no se necesiten (pasado el periodo de retención), utilizando el método de destrucción adecuado según el tipo de confidencialidad del informe, y que mientras tanto, sean almacenados en lugar seguro.

3.5.1.9 Controles sobre las salidas on-line

Intentan garantizar que un usuario no autorizado no pueda interceptar las líneas de comunicaciones, por lo que es necesario disponer de líneas seguras, desde el punto de vista físico.

Además deben de garantizar que los informes por pantalla sólo estén accesibles al personal autorizado.

3.5.2 Controles sobre ficheros

Los objetivos de estos controles son evitar sobreescribir o borrar información válida.

En el propio fichero se puede grabar la identificación del fichero, el número de generación del fichero, la fecha de retención y los totales de control del fichero, entre otros.

3.5.3 Consideraciones de Efectividad / Eficiencia

La efectividad de un sistema depende en gran medida de la calidad de la información que proporciona. Esta información debe de ser precisa, adecuada, fiable, y estar obtenida a tiempo. El método de presentación (medio, estilo, diseño, etc.) es asimismo muy importante, así como los tiempos de respuesta en sistemas interactivos.

La eficiencia se logra al eliminar informes inútiles, páginas o líneas no necesarias (con el consiguiente ahorro de papel), y con un diseño correcto de los informes.

3.5.4 Resumen

Los objetivos de los controles de salida sobre los informes intentan preservar la integridad de los datos y ayudar a alcanzar efectividad y eficiencia.

Además, los controles sobre ficheros sirven para proteger la privacidad de los datos y para garantizar que los ficheros no se modifican o borran por error o a propósito antes de que termine su período de vida útil.

La calidad de los informes es muy importante para que alcancen su cometido: una toma de decisiones adecuada.

3.6 Controles de Pistas de Auditoria.

Las Pistas de Auditoría son una lista cronológica de eventos que le han ocurrido a una entidad. Podemos distinguir dos tipos de pistas de auditoría: de contabilidad y de operaciones.

Las primeras muestran las operaciones realizadas sobre los ítems de datos de una Base de Datos, mientras que las segundas recogen los eventos ocurridos durante la ejecución de una aplicación.

3.6.1 Pistas de Auditoría de Contabilidad

Permiten que una transacción se pueda seguir desde su origen, a través de los ítems de datos sobre los que opera (proceso conocido como Implosión), o permiten la reconstrucción en el tiempo de las series de operaciones realizadas sobre los ítems de datos (Explosión)

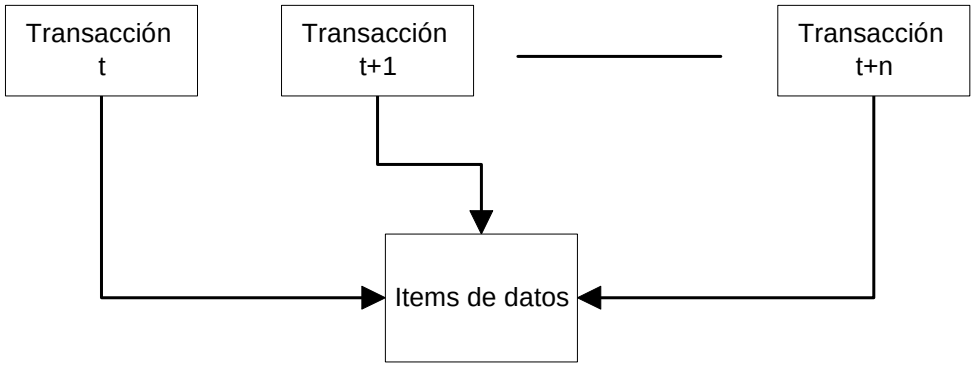


Fig. 3.6 Proceso de Implosión

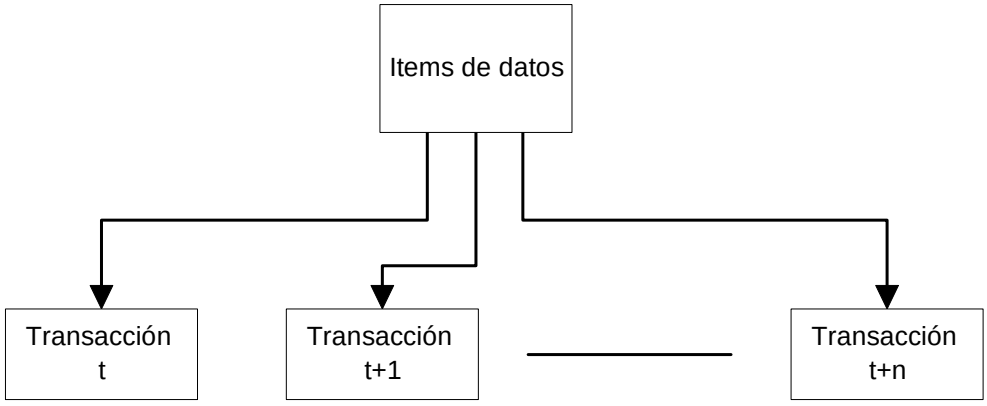


Fig. 3.7 Proceso de Explosión

3.6.1.1 Requerimientos operativos de las Pistas de Auditoría de Contabilidad

Se necesita software que permita realizar cuatro tipos de operaciones sobre una pista de auditoría de contabilidad: creación, modificación, borrado y recuperación.

Como la mayoría del software generalizado soporta este tipo de operaciones, normalmente no es necesario un software específico de auditoría.

3.6.1.1.1 Creación

Existen dos tipos de registros de pistas de auditoría: permanentes y temporales. Por ejemplo, una operación que actualiza un saldo se puede guardar como un registro permanente, mientras que si el auditor quiere seguir una determinada transacción durante un cierto tiempo, se puede crear un registro temporal con la información necesaria.

3.6.1.1.2 Modificación.

Normalmente no se hace a menos que las aplicaciones realicen procesos erróneos (con lo que las pistas serán erróneas) o que las subrutinas que crean las pistas de auditoría estén mal codificadas.

3.6.1.1.3 Recuperación.

No difiere de la de cualquier otro tipo de registro de una Base de Datos.

3.6.1.1.4 Borrado.

Se debe de hacer periódicamente para no aumentar demasiado el volumen de datos.

3.6.2 Pistas de Auditoría de Operaciones.

Normalmente son difíciles de usar y de entender. Dado que la mayoría de los sistemas ya tienen un LOG para el seguimiento de las transacciones, se debe estudiar cuidadosamente si vale la pena establecerlas, ya que sin duda van a aumentar la carga del sistema.

3.7 Controles de backup y recuperación en Bases de Datos

Son necesarios para restablecer la existencia de la BD física en caso de pérdida total o parcial de la misma.

El auditor debe de estar familiarizado con todos los temas relacionados con la seguridad de las bases de datos: LOG’s, COMMIT, ROLLBACK, transacciones atómicas, etc.

3.7.1 Estrategias de Copias de Seguridad y Recuperación

Existen distintas estrategias para la obtención de copias de seguridad de la información. Las mas corrientes se indican a continuación.

1. Guardar tres (o más) generaciones de un mismo fichero, reutilizándose el soporte una vez alcanzada la última generación.

2. Grabación dual. Mantener dos copias separadas de la Base de Datos y actualizarlas simultáneamente.

3. Volcado o vaciado. Consiste en copiar la BD sobre un medio de backup (cintas, cartuchos), para poder recuperarla cuando sea necesario.

4. Logging. Consiste en grabar la transacción que modifica la BD, una imagen del registro cambiado o los parámetros de cambio que resultan de la modificación.

3.7.2 Ejercicios y casos

3.7.2.1 Control de informes

Ubend, Inc. es una empresa que distribuye piezas de fontanería con sede en Sidney, con tiendas en las ciudades principales del país. Cada tienda envía sus transacciones a la Central para que sean procesadas, y ésta les devuelve microfichas con la totalidad de las transacciones procesadas, así como un informe resumido, para que sean utilizadas por las tiendas para su contabilidad.

Formas parte del equipo de Auditoría externa que se encuentra examinando los controles sobre las salidas en el Sistema de Proceso de Datos de la Central, y tu jefe ha preparado una lista de los siguientes objetivos de la auditoría:

a.) Garantizar que los informes no se puedan perder.
b.) Garantizar que los informes no se puedan robar.
c.) Garantizar que personas no autorizadas puedan acceder a los informes.
d.) Garantizar que los informes se guardan durante siete años, debido a requerimientos de Hacienda.

Se pide:

Un informe indicando las diferencias en los controles para las microfichas y los informes, con el fin de alcanzar los objetivos de la auditoría.

3.7.2.2 Informes con colores

Rosendale Savings and Loan acaba de instalar un Sistema de Información de Gestión para realizar operaciones de préstamos. Cuando un cliente solicita un préstamo, un administrativo utiliza un terminal para pedir información sobre el estado de finanzas del cliente en la Base de Datos de la empresa.

La empresa ha adquirido terminales gráficos en color para utilizar con el nuevo sistema, y destacar así diferente información de interés: por ejemplo, es posible detectar un saldo negativo porque figura en rojo y uno positivo porque aparece en verde. Se utilizan también gráficos en color para mostrar tendencias en los saldos durante los últimos cinco años.

Dado que se han producido algunos errores en la toma de decisiones basada en la información proporcionada por el nuevo sistema, éste ha sido comprobado cuidadosamente, y no se han apreciado errores en la lógica de proceso.

Se pide:

Escribir un informe con sugerencias sobre lo que puede estar fallando, indicando las acciones que se deben de tomar para corregir el problema.

3.8 Bibliografía Temas 1 al 3

[Boritz, 1979] Boritz, J. Effrim. “Computer Guide’ 79”.

[Bright & Enison, 1976] Bright, Herbert and R.L. Enison. “Cryptography Using Modular Software Elements”. Proceedings of the 1976 National Computer Conference.

[Canadian, 1970] Canadian Institute of Chartered Accountants (1970)

[Davis et al, 1981], Davis, Gordon B., Donald Adams and Carol A. Schaller. “Auditing an EDP”. Institute of Certified Public Accountants.

[EDP, 1977] EDP, “Control Objetives”. Auditors Foundation for Education & Research (1977)

[Ehrsam et al, 1978] Ehrsam, W.F., S.M. Matyas, C.H. Meyer and W.L. Tuchman. “A Cryptographic Key Management System for Implementing the Data Encryption Standard”. IBM Systems Journal.

[Lennon, 1978]

[Lucas, 1975] Lucas, Henry. Towards Creative System Design.

[Miller, 1956] Miller, George A. “The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capability for Processing Information”. The Psychological Review.

[Miller, 1973] Miller, Curt. “”Union Dime Picks Up the Pieces in $1.5 Million Embezzlement Case”

[Owsowitz & Sweetland, 1965] Owsowitz, S. and A. Sweetland. “Factors Affecting Coding Errors”. The Rand Corporation.

[Parker, 1976] Parker, Donn B. Crime by Computer. N.Y.

[Perry, 1977] Perry, William E. “Carrer Advancement for EDP Auditor”. EDPACS.

[Rittemberg, 1977] Rittemberg, Larry E. “Auditor Independence and Systems Design”. The Institute of Internal Auditors.

[Stepczyk, 1974] Stepczyk, S.M. “Requirements for Secure Operating Systems”. TRW Systems.

[Weiss, 1977] Weiss, Harold. “EDP Audit Job Descriptions”. EDPACS.