**12.1. ¿Cuál es la diferencia entre un campo y un registro?**

Un **campo** es el elemento básico de los datos. Un campo individual contiene un único valor, tal como el apellido de un empleado, una fecha o el valor de un sensor. Se caracteriza por su longitud y el tipo de datos (por ejemplo, ASCII, cadena de caracteres, decimal). Dependiendo del diseño del fichero, el campo puede tener una longitud fija o variable. En este último caso, el campo está formado normalmente por dos o tres subcampos: el valor real almacenado, el nombre del campo, y en algunos casos, la longitud del campo. En otros casos de campos de longitud variable, la longitud del campo se indica mediante el uso de símbolos de demarcación especiales entre campos.

Un **registro** es una colección de campos relacionados que pueden tratarse como una unidad por alguna aplicación. Por ejemplo, un registro de empleado podría contener campos tales como nombre, número de seguridad social, clasificación de trabajo, fecha de contratación, etc. De nuevo, dependiendo del diseño, los registros pueden ser de longitud fija o variable. Un registro tendrá una longitud variable si alguno de sus campos tiene longitud variable o si el número de campos puede variar. En este último caso, cada campo se acompaña normalmente de un nombre de campo. En cualquier caso, el registro completo incluye normalmente un campo longitud.

**12.2. ¿Cuál es la diferencia entre un fichero y una base de datos?**  
  
Un **fichero** es una colección de campos similares. El fichero se trata como una entidad única por parte de los usuarios y las aplicaciones. Los ficheros se pueden referenciar por nombre. Dichos ficheros se pueden crear y borrar. Las restricciones de control de acceso normalmente se aplican a nivel del fichero. Es decir, en un sistema compartido, el acceso a los ficheros completos es permitido o denegado a los usuarios y los programas. En algunos sistemas más sofisticados, tales controles se realizan a nivel de registro o incluso a nivel de campo. **¿Un fichero no sería un conjunto de registros similares?**  
  
Una **base de datos** es una colección de datos relacionados. Los aspectos esenciales de una base de datos son que la relación que exista entre los elementos de datos sea explícita y que la base de datos se diseña para su uso por parte de varias aplicaciones diferentes. Una base de datos podría contener toda la información relacionada con una organización o proyecto, tal como información de negocio o de estudio científico. La base de datos está formada por uno o más tipos de ficheros. Normalmente, hay un sistema de gestión de base de datos separado del sistema operativo, aunque hace uso de algunos programas de gestión de ficheros.

**12.3. ¿Qué es un sistema de gestión de ficheros?**

Un sistema de gestión de ficheros es aquel conjunto de software de sistema que proporciona servicios a los usuarios y aplicaciones en el uso de ficheros. Típicamente, la única forma en la que un usuario o aplicación puede acceder a los ficheros es a través del sistema de gestión de ficheros. Esto elimina la necesidad de que el usuario o programador desarrolle software de propósito especial para cada aplicación. Además, proporciona al sistema una forma consistente y bien definida de controlar su recurso más importante.

**12.4. ¿Qué criterios son importantes a la hora de escoger una organización de ficheros?**

Se utiliza el término organización de fichero para referirse a la estructura lógica de los registros determinados por la forma en la que se acceden. La organización física del fichero en almacenamiento secundario depende de la estrategia de bloques y de asignación de ficheros.

Para escoger una organización de ficheros, son importantes varios criterios:

• Tiempo de acceso corto.

• Facilidad de actualización.

• Economía de almacenamiento.

• Mantenimiento sencillo.

• Fiabilidad.

**12.5. Liste y defina brevemente cinco organizaciones de ficheros.**   
Se describen cinco organizaciones fundamentales. La mayoría de las estructuras utilizadas en los sistemas reales cae dentro de una de estas categorías y se puede implementar con una combinación de estas organizaciones. Las cinco organizaciones son:

• La **pila**. La forma menos complicada de organización de ficheros se puede denominar pila. Los datos se almacenan en el orden en el que llegan. Cada registro está formado por un conjunto de datos. El propósito de la pila es simplemente acumular la masa de datos y guardarlos. Los registros podrían tener diferentes campos o similares campos en diferentes órdenes.

• El **fichero secuencial**.La forma más común de estructura de fichero es el fichero secuencial. En este tipo de ficheros, se utiliza un formato fijo para los registros. Todos los registros son de igual tamaño y están compuestos por el mismo número de campos de longitud fija en un orden específico. Debido a que la longitud y la posición de cada campo son conocidas, sólo se necesita almacenar los valores de los campos; el nombre y longitud de cada campo son atributos de la estructura del fichero.

Un campo particular, normalmente el primer campo de cada registro, se denomina **campo clave**. El campo clave identifica de forma única el registro; por tanto, los valores de la clave de diferentes registros son siempre diferentes.

• El **fichero secuencial indexado**. Una técnica popular para eliminar las desventajas del fichero secuencial es utilizar los ficheros secuenciales indexados. El fichero secuencial indexado mantiene las características clave del fichero secuencial: los registros se organizan en secuencia, basándose en un campo clave. Dos características se añaden: un índice al fichero que da soporte al acceso aleatorio y un fichero de desbordamiento. El índice proporciona una capacidad de búsqueda para alcanzar rápidamente la vecindad de un registro deseado. El fichero de desbordamiento es similar al fichero registro, utilizado con un fichero secuencial, pero se integra de tal forma que un registro en el fichero de desbordamiento se localiza mediante un puntero desde su registro predecesor.

• El **fichero indexado**. Los ficheros secuenciales tradicionales solo permiten búsquedas basadas en un único campo clave, lo que es limitado si se necesita buscar por otros atributos. El fichero secuencial indexado introduce múltiples índices para diferentes campos, lo que permite una mayor flexibilidad en las búsquedas. Utiliza múltiples índices, cada uno asociado con un campo que puede ser sujeto a búsqueda. Abandona los conceptos de secuencialidad y clave única, permitiendo el acceso a registros solo a través de los índices.

• El fichero de acceso directo o ***hash****:*

- **Acceso Rápido:** Utiliza la capacidad de los discos para acceder directamente a cualquier bloque de datos si se conoce su dirección.

- **Clave Necesaria:** Cada registro necesita una clave para calcular su ubicación en el disco mediante una función hash.

- **Sin Ordenación Secuencial:** No hay un orden secuencial de los registros; la función hash distribuye las claves a direcciones específicas en el disco. Recordemos que en la **ordenación secuencial,** los registros en un fichero se almacenan en un orden específico, generalmente alfabético, numérico o cronológico, basado en el valor de uno o más campos clave.

**- Función Hash:** Convierte la clave del registro en una dirección en el disco, permitiendo un acceso directo y rápido.

**12.6. ¿Por qué es menor el tiempo de búsqueda medio de un registro en un fichero para un fichero secuencial indexado que para un fichero secuencial?**

El menor tiempo de búsqueda medio en un fichero secuencial indexado se debe a que los índices permiten acceder directamente a los registros deseados, eliminando la necesidad de realizar una búsqueda secuencial completa a través de todos los registros. Esto hace que las búsquedas sean mucho más rápidas y eficientes.

**12.7. ¿Cuáles son las operaciones típicas que se pueden realizar sobre un directorio?**

**Directorios:** El directorio contiene información sobre los ficheros, incluyendo atributos, ubicación y propiedad. Gran parte de esta información, especialmente la que concierne a almacenamiento, la gestiona el sistema operativo. El directorio es a su vez un fichero, accesible por varias rutinas de gestión de ficheros. Aunque parte de la información de los directorios está disponible para los usuarios y las aplicaciones, esto se proporciona generalmente de forma indirecta por las rutinas del sistema.

Para comprender los requisitos de la estructura de un fichero, es útil considerar los tipos de operaciones que se pueden llevar a cabo sobre directorios:

**• Buscar.** Cuando un usuario o aplicación referencia un fichero, el directorio debe permitir encontrar la entrada correspondiente a dicho fichero.

**• Crear fichero.** Cuando se crea un nuevo fichero, se debe añadir una entrada al directorio.

**• Borrar fichero.** Cuando se borra un fichero, se debe eliminar una entrada del directorio.

**• Listar directorio.** Se puede solicitar ver el directorio completo o una porción del mismo. Generalmente, el usuario solicita esta petición y como resultado obtiene un listado de todos los

ficheros de los cuales es propietario, más algunos de los atributos de cada fichero (por ejemplo, información de control de acceso, información de uso).

**• Actualizar directorio.** Debido a que algunos atributos se almacenan en el directorio, un cambio en uno de estos atributos requiere un cambio en la entrada de directorio correspondiente.

12.8. ¿Cuál es la relación entre una ruta de un fichero y un directorio de trabajo?

**Ruta de un Fichero 🡪 utiliza rutas absolutas:**

* **Definición:** La ruta de un fichero es el camino completo que se sigue desde el directorio raíz o maestro hasta el fichero específico. Este camino se compone de una secuencia de nombres de directorios que finalizan con el nombre del fichero.
* **Ejemplo:** En el sistema de archivos, una ruta completa puede ser algo como /Usuario\_B/Texto/Unidad\_A/ABC, donde cada nombre está separado por una barra /.

**Directorio de Trabajo 🡪 utiliza rutas relativas:**

* **Definición:** El directorio de trabajo es el directorio actual en el que un usuario o proceso se encuentra operando. Este directorio actúa como un punto de referencia para las operaciones de archivos y permite a los usuarios referenciar archivos de manera más simple, sin necesidad de escribir la ruta completa.
* **Cambio de Directorio:** Los usuarios pueden cambiar su directorio de trabajo durante la sesión para facilitar el acceso a diferentes ficheros dentro del sistema de archivos.

**12.9. ¿Cuáles son los derechos de acceso típicos que se pueden conceder o denegar a un usuario particular sobre un fichero particular?**

En un sistema multiusuario, existe casi siempre el requisito de permitir que los ficheros se compartan entre varios usuarios. Existen dos aspectos relacionados: derechos de acceso y la gestión de acceso simultáneo.

**DERECHOS DE ACCESO**

El sistema de ficheros debería proporcionar una herramienta flexible para permitir la compartición de ficheros extensiva entre los usuarios. El sistema de ficheros debería proporcionar varias opciones de tal forma que el acceso a un fichero particular se pueda controlar. Típicamente, a los usuarios o grupos de usuarios se les concede ciertos derechos de acceso a un fichero. Se ha utilizado un amplio rango de derechos de acceso. La siguiente lista es representativa de los derechos de acceso que se asignan a un usuario particular para un determinado fichero:

**• Ninguno.** El usuario no puede incluso conocer la existencia del fichero, y por tanto, tampoco

puede acceder a él. Para forzar esta restricción, el usuario no tiene permiso de lectura del directorio en el cual se incluye este fichero.

**• Conocimiento.** El usuario puede determinar si el fichero existe y quién es su propietario. El

usuario entonces es capaz de solicitar al propietario derechos de acceso adicionales.

**• Ejecución.** El usuario puede cargar y ejecutar un programa pero no copiarlo. Los programas

propietarios utilizan normalmente estas restricciones.

**• Lectura.** El usuario puede leer el fichero para cualquier propósito, incluyendo copia y ejecución. Algunos sistemas son capaces de forzar una distinción entre ver y copiar. En el primer caso, el usuario puede ver el contenido del fichero, pero no puede realizar una copia.

**• Adición.** El usuario puede añadir datos al fichero, frecuentemente sólo al final, pero no puede

modificar o borrar cualquiera de los contenidos del fichero. Este derecho es útil para recolectar datos de varias fuentes.

**• Actualización.** El usuario puede modificar, borrar o añadir datos al fichero. Esto normalmente

incluye escribir el fichero al inicio, reescribirlo completa o parcialmente y borrar todos o una

porción de los datos. Algunos sistemas diferencian entre distintos grados de actualización.

**• Cambio de protección.** El usuario puede cambiar los derechos de acceso otorgados a otros

usuarios. Normalmente, sólo el propietario del fichero goza de este derecho. En algunos sistemas, el propietario puede extender este derecho a otros. Para prevenir del abuso de este mecanismo, el propietario del fichero normalmente puede especificar qué derechos podría cambiar un usuario con este tipo de permiso.

**• Borrado.** El usuario puede borrar el fichero del sistema de ficheros.

Estos derechos se pueden considerar como una jerarquía, con cada uno de los derechos conteniendo a aquellos que le preceden. Por tanto, si un usuario particular tiene el permiso de actualización para un fichero concreto, dicho usuario también tendrá los siguientes derechos: conocimiento, ejecución, lectura y adición.

**12.10. Liste y defina brevemente tres tipos de bloques utilizados.**

Los registros son las unidades lógicas de acceso de un fichero estructurado, mientras que los bloques son las unidades de E/S con almacenamiento secundario. Para que la E/S se pueda realizar, los registros se deben organizar como bloques.

La preocupación viene por el hecho de que bloques más grandes requieren *buffers* de E/S mayores, haciendo la gestión de *buffers* más difícil.

Dado el tamaño de un bloque, se pueden utilizar tres métodos:

**• Bloques fijos.** Se utilizan registros de longitud fija y se almacenan en un bloque un número

integral de registros. Podría haber espacio no utilizado al final de cada bloque. Esto se denomina fragmentación interna.

**• Bloques expandidos de longitud variable.** Se utilizan registros de longitud variable y se

empaquetan en bloques sin dejar espacio no utilizado. Por tanto, algunos registros deben expandirse a lo largo de dos bloques, con su continuación indicada por un puntero al bloque

sucesor.

**• Bloques no expandidos de longitud variable.** Se utilizan registros de longitud variable, pero

no se emplea expansión. Hay espacio malgastado en la mayoría de los bloques debido a la incapacidad para utilizar el resto de un bloque si el siguiente registro es mayor que el espacio no

utilizado restante.

**12.11. Liste y defina brevemente tres métodos de asignación de ficheros.**

**Asignación Contigua:**

* **Descripción:** En la asignación contigua, se asigna un conjunto contiguo de bloques en el momento de la creación del fichero. Esto significa que todos los bloques que forman un fichero están físicamente adyacentes en el disco.
* **Ventajas:**
  + Mejor rendimiento de E/S en el procesamiento secuencial, ya que múltiples bloques se pueden leer de una vez.
  + Fácil acceso a un bloque específico, ya que su ubicación se calcula fácilmente.
* **Desventajas:**
  + Fragmentación externa, lo que dificulta encontrar bloques contiguos libres de longitud suficiente.
  + Necesidad de predecir el tamaño del fichero al crearlo, lo que puede llevar a una asignación ineficiente de espacio.
  + Requiere ocasionalmente un algoritmo de compactación para liberar espacio.

**Asignación Encadenada:**

* **Descripción:** En la asignación encadenada, los bloques de un fichero no están necesariamente contiguos. Cada bloque contiene un puntero al siguiente bloque de la cadena.
* **Ventajas:**
  + No hay fragmentación externa, ya que se asigna un bloque a la vez según sea necesario.
  + No es necesario predecir el tamaño del fichero al crearlo.
* **Desventajas:**
  + El acceso a un bloque específico requiere seguir la cadena desde el principio, lo que puede ser lento.
  + Puede causar problemas de rendimiento debido a la falta de proximidad, ya que los bloques pueden estar dispersos en el disco.
  + Traer varios bloques a la vez requiere múltiples accesos al disco.

**Asignación Indexada:**

* **Descripción:** En la asignación indexada, se usa un índice separado para cada fichero. El índice contiene una entrada por cada porción asignada al fichero. El índice de ficheros se guarda en un bloque separado y la entrada para el fichero en la tabla de asignación de ficheros apunta a dicho bloque.
* **Ventajas:**
  + Elimina la fragmentación externa (cuando se usa asignación por bloques de tamaño fijo).
  + Mejora la proximidad (cuando se usa asignación por porciones de tamaño variable).
  + Soporta tanto acceso secuencial como directo a los ficheros.
* **Desventajas:**
  + Puede requerir una consolidación de ficheros periódica para reducir el tamaño del índice.
  + El índice mismo ocupa espacio adicional en el disco.

**Explicación del ChatGPT:**  
Vamos a desglosar los conceptos de "punteros" y "bloques" para que sea más claro:

### Conceptos Básicos

- \*\*Bloques\*\*:

- Los discos duros y otros dispositivos de almacenamiento dividen su espacio en unidades más pequeñas llamadas bloques. Un bloque es la unidad mínima de almacenamiento en un disco. Por ejemplo, un archivo puede estar compuesto por varios bloques.

- \*\*Punteros\*\*:

- Un puntero es una referencia a otra ubicación en el disco. En el contexto de asignación de archivos, un puntero indica dónde está almacenado el siguiente bloque de un archivo.

### Métodos de Asignación de Archivos

#### 1. Asignación Contigua

- \*\*Descripción\*\*:

- Imagina que tienes un archivo que ocupa varios bloques. En la asignación contigua, todos estos bloques se guardan uno tras otro en el disco.

- \*\*Ejemplo Visual\*\*:

- Si un archivo necesita 3 bloques y empieza en el bloque 5, se guardará en los bloques 5, 6 y 7.

#### 2. Asignación Encadenada

- \*\*Descripción\*\*:

- Aquí, los bloques del archivo no tienen que estar uno al lado del otro. En su lugar, cada bloque contiene un puntero que indica dónde está el siguiente bloque del archivo.

- \*\*Ejemplo Visual\*\*:

- Supongamos que el archivo empieza en el bloque 5, el bloque 5 tiene un puntero que dice "el siguiente bloque está en el bloque 9", y el bloque 9 tiene un puntero que dice "el siguiente bloque está en el bloque 12", y así sucesivamente.

#### 3. Asignación Indexada

- \*\*Descripción\*\*:

- En este método, hay una estructura especial llamada índice que guarda punteros a todos los bloques del archivo. El índice mismo está guardado en un bloque separado.

- \*\*Ejemplo Visual\*\*:

- Piensa en el índice como una lista. Si un archivo tiene bloques en 5, 9, y 12, el índice tendría punteros a estos bloques. El archivo tiene una referencia a este índice.

### Visualización Simplificada

Para entender mejor, imagina un archivo como un libro:

- \*\*Asignación Contigua\*\*: Todas las páginas del libro están una después de la otra en orden.

- \*\*Asignación Encadenada\*\*: Cada página del libro tiene una nota al final que dice dónde está la siguiente página.

- \*\*Asignación Indexada\*\*: Hay un índice al inicio del libro que te dice en qué páginas están los diferentes capítulos o secciones.