Principios de Sistemas Operativos

Interfaz de Archivos

Armando Arce, Escuela de Computación, arce@itcr.ac.cr

Tecnológico de Costa Rica

Interfaz del sistema de archivos

Generalmente, el sistema de archivos es el aspecto más visible de un sistema operativo.

- Este proporciona mecanismos para el almacenamiento en línea de datos y programas del mismo sistema y de todos los usuarios; así como para el acceso a esos datos y programas.
- El sistema de archivo está compuesto de dos diferentes componentes: una colección de archivos, cada uno de los cuales almacena una serie de datos relacionados, y una estructura de directorios, que organiza todos los archivos del sistema y proporciona información acerca de los mismos.

Concepto de archivo

El sistema operativo realiza una abstracción de las propiedades físicas de los diferentes dispositivos de almacenamiento, con el fin de definir una unidad lógica de almacenamiento, *el archivo*.

- Los archivos son mapeados por el sistema operativo sobre los dispositivos físicos.
- Un archivo es una colección de información relacionada, con un nombre, que se graba en almacenamiento secundario.
- En general, un archivo es una secuencia de bytes, líneas o registros cuyo significado está definido por el creador y el usuario del mismo.
- Por tanto, el concepto de archivo es extremadamente general.

Atributos de archivo

Los atributos de un archivo varían de un sistema a otro, pero generalmente son los siguientes:

- Nombre: El nombre simbólico del archivo.
- Identificador: Identifica al archivo internamente.
- Tipo: Algunos sistemas soportan diferentes tipos de archivos.
- Ubicación: Esta información es un puntero a un dispositivo y a la ubicación del archivo.
- Tamaño: El tamaño actual del archivo, y posiblemente, el tamaño máximo.
- Protección: Información de control acceso (leer,escribir,ejecutar)
- Fecha, hora, identificación de usuario: Estos datos son útiles para protección, seguridad y monitorización de uso

Atributos de archivo

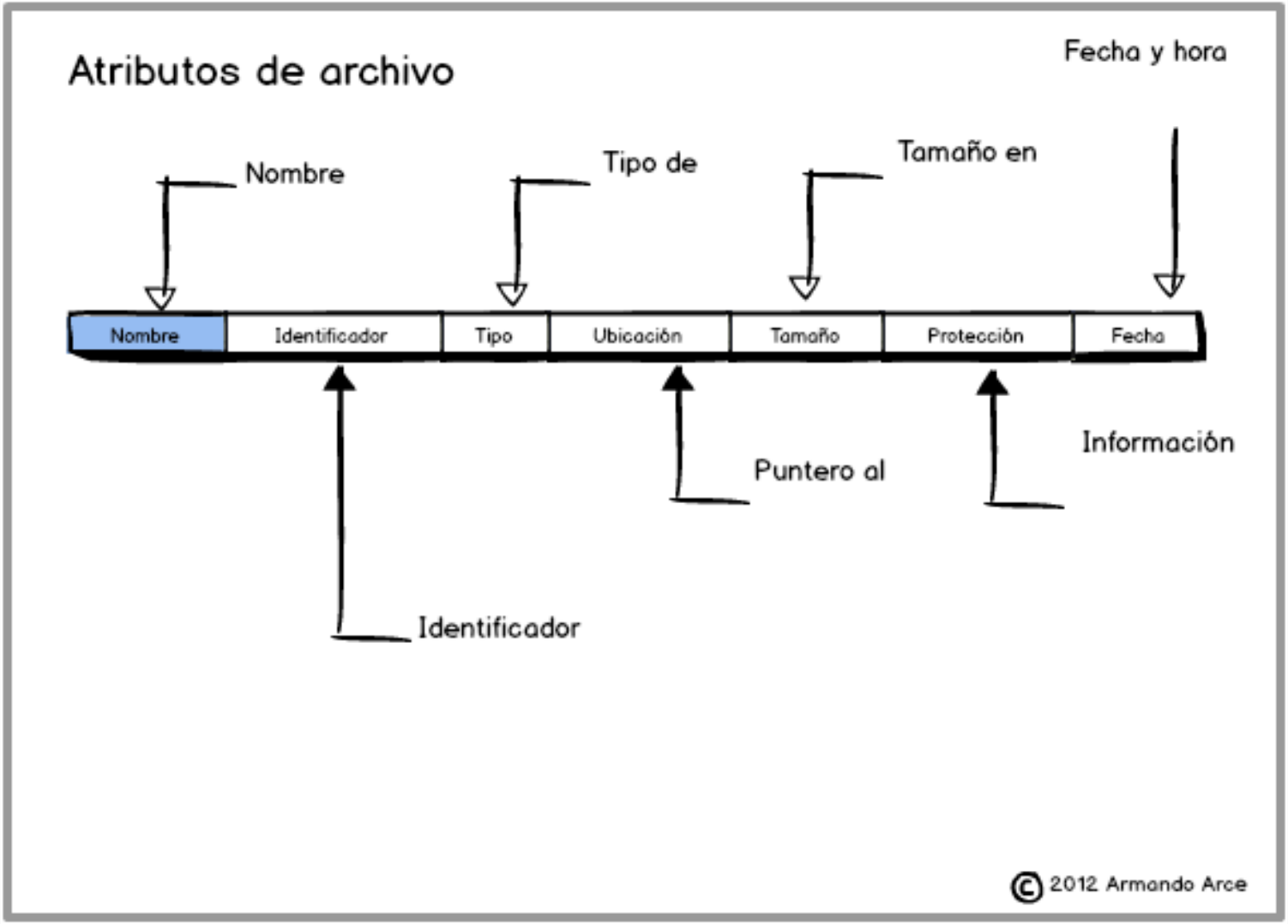

Figure 1:

Atributos de archivo

La información acerca de los archivos se almacena en la estructura de directorios, que también reside en el almacenamiento secundario.

- Una entrada de directorio está compuesta del nombre del archivo y su identificador.
- El identificador permite localizar los demás atributos del archivo.

Operaciones con los archivos

Algunos sistemas abren implícitamente los archivos cuando se realiza la primera referencia a los mismos. El archivo se cerrará automáticamente cuando termine el trabajo que lo abrió.

- Sin embargo, la mayoría de los sistemas requieren que el programador abra el archivo explícitamente con la llamada al sistema *open()*.
- Esta operación toma un nombre de archivo y explora el directorio, copiando la entrada de directorio en la *tabla de archivos abiertos*.
- La llamada retorna un puntero que referencia la entrada respectiva dentro de la tabla de archivos abiertos.
- Es ese puntero, y no el nombre simbólico, lo que se utiliza en las operaciones de E/S, evitando nuevas búsquedas en directorios.

Operaciones con los archivos

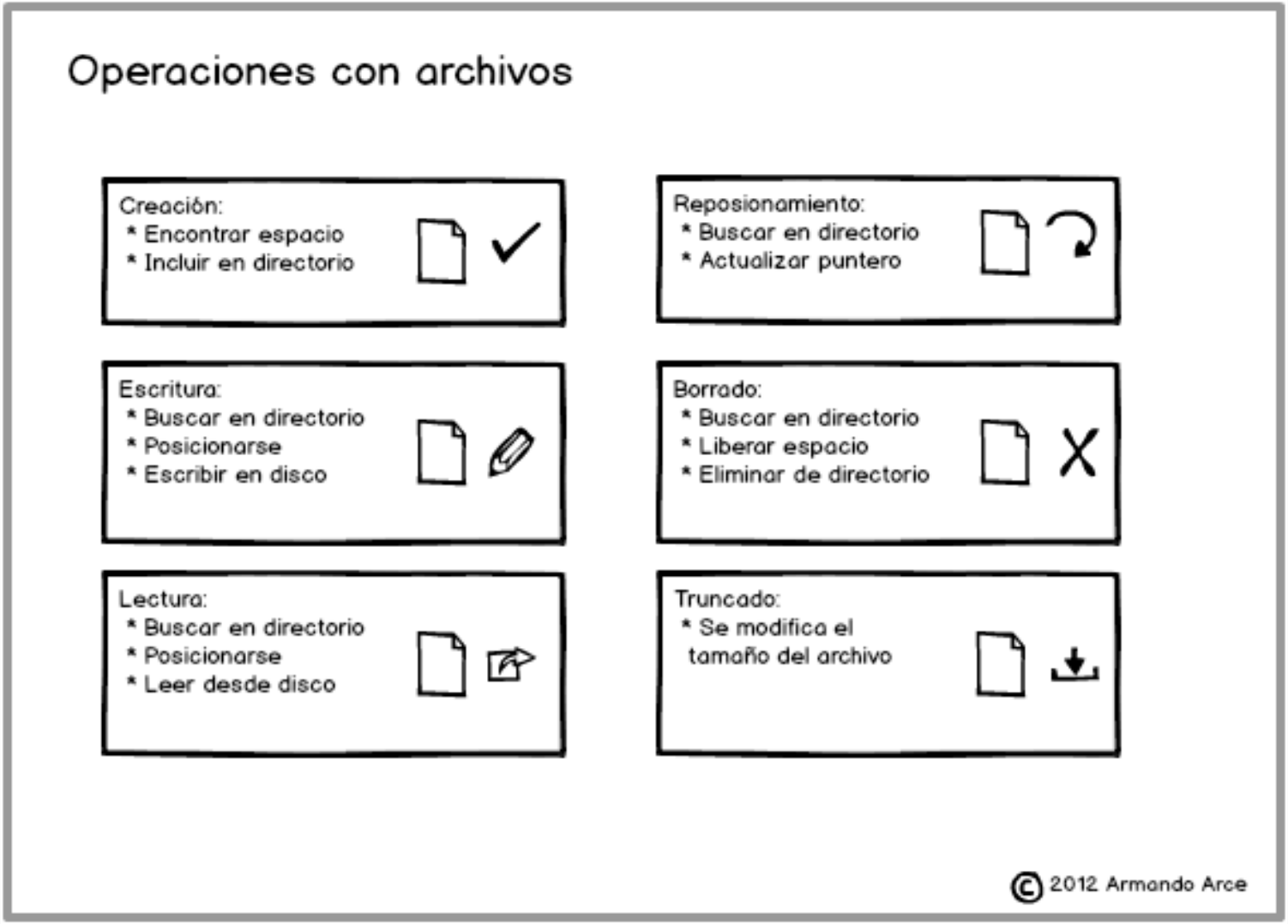

Figure 2:

Operaciones con los archivos

El sistema operativo utiliza dos niveles de tablas internas: un tabla por cada proceso y una tabla global del sistema.

- La *tabla de archivos abiertos* de cada proceso indica todos los archivos que ese proceso ha abierto.
- En la *tabla global* contiene información independiente del proceso.
- Una vez que un proceso ha abierto un archivo, se incluye una entrada para dicho archivo en la tabla global.

Operaciones con los archivos

Con cada archivo abierto se asocian diferentes tipos de datos:

- Puntero de archivo: el sistema deberá registrar la ubicación correspondiente a la última lectura-escritura, utilizando un puntero de posición actual dentro de cada archivo
- Contador de aperturas del archivo: este contador controla el número de aperturas y cierres y alcanzará el valor cero cuando el último proceso cierre el archivo y así podrá eliminar la entrada
- Ubicación del archivo dentro del disco: la información para ubicar el archivo en el disco se almacena en memoria
- Derechos de acceso: permite al sistema operativo autorizar o denegar las solicitudes de E/S

Operaciones con los archivos

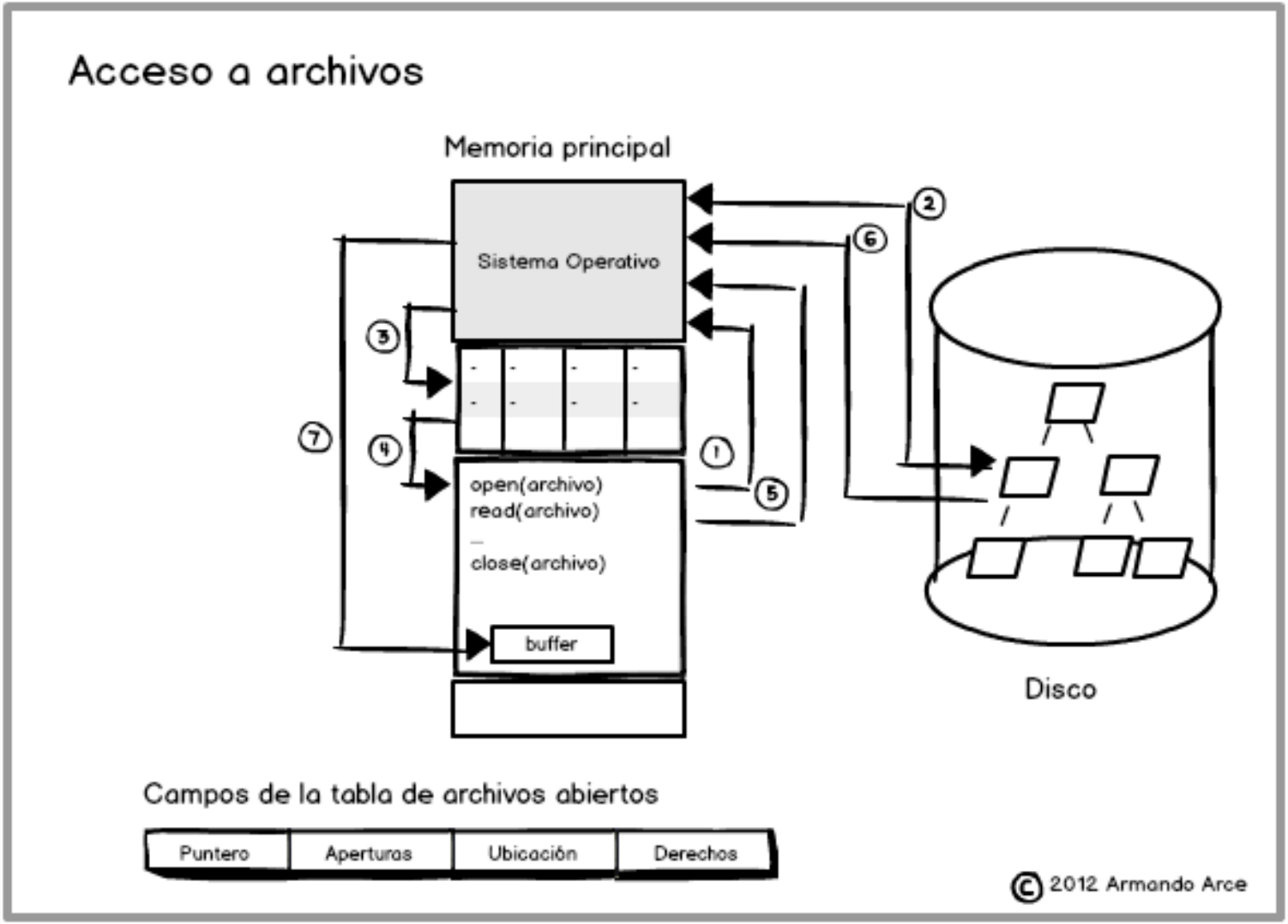

Figure 3:

Operaciones con los archivos

El sistema operativo proporciona funciones para bloquear un archivo o determinadas secciones de un archivo.

- Los *bloqueos* permiten que un proceso bloquee un archivo e impida que otros procesos puedan acceder al mismo.
- Existen dos tipos de bloqueos: el *bloqueo compartido* es similar a un bloqueo lector, es decir, varios procesos pueden adquirir bloqueos concurrentemente; el *bloqueo exclusivo* se comporta como un bloqueo escritor, sólo puede adquirir dicho tipo de bloqueo un proceso cada vez.

Operaciones con los archivos

Además, los sistemas operativos pueden proporcionar mecanismos de bloqueo de archivos obligatorios o sugeridos.

- Si el *bloqueo es obligatorio* (p.ej. Windows), después que un proceso adquiera un bloqueo, el sistema impedirá a los demás procesos el acceso al archivo bloqueado.
- Esto sucede aún cuando el proceso no haya solicitado el bloqueo.
- Si el *bloqueo es de tipo sugerido* (p.ej. Unix), el sistema operativo no impedirá a los procesos adquirir el acceso.
- Son los procesos los que deben solicitar el bloqueo explícitamente.

Tipos de archivo

Cuando se diseña un sistema de archivos siempre se debe considerar si se debe reconocer y soportar el concepto de tipo de archivo.

- Si un sistema reconoce el tipo de archivo, podrá operar con ese archivo de diferentes formas.
- Por ejemplo, un error muy común es tratar de imprimir un archivo binario, una operación que sí tendría sentido con un archivo de texto.

Tipos de archivo

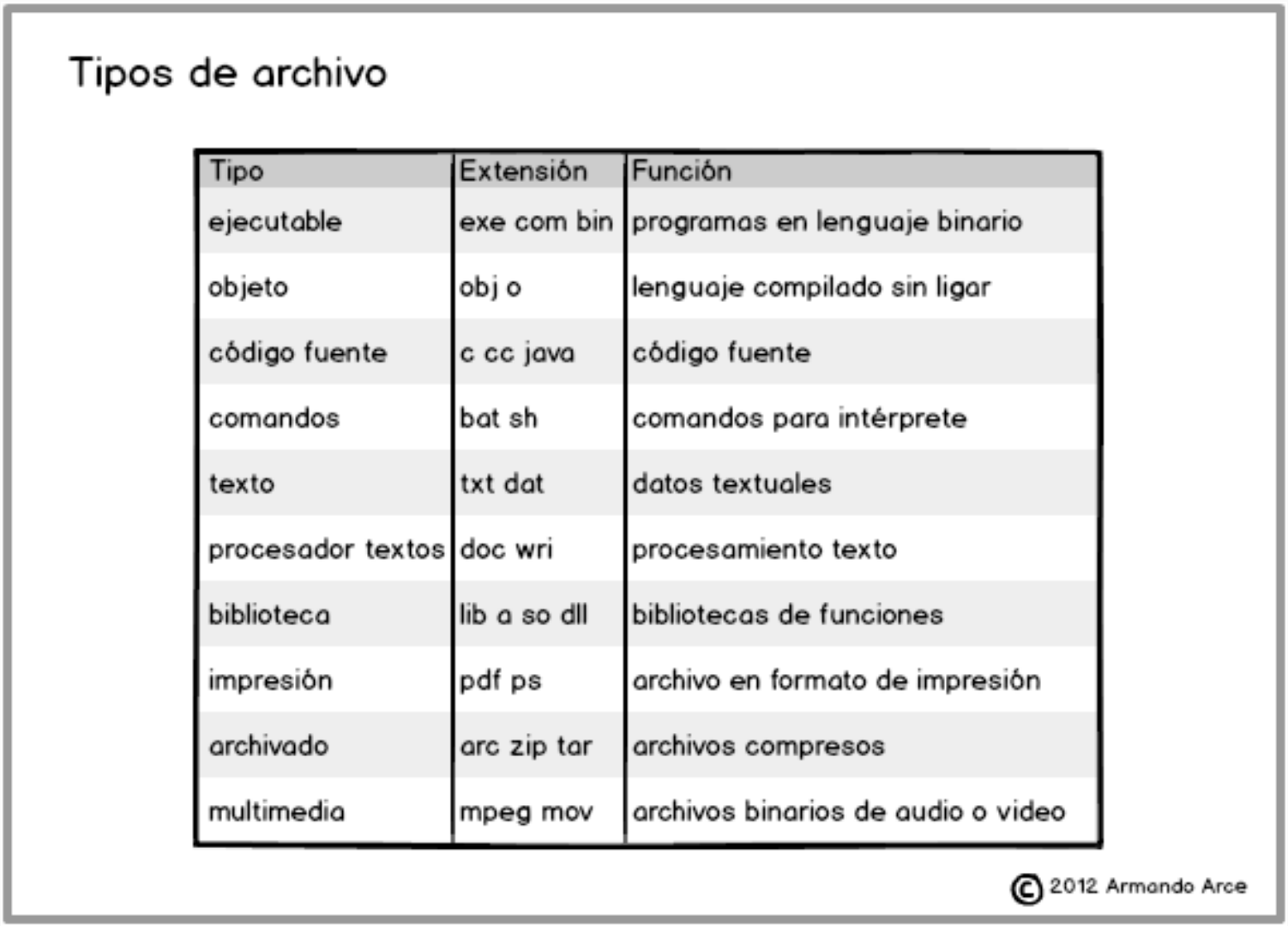

Figure 4:

Tipos de archivo

Una técnica común para implementar los tipos de archivo consiste en incluir el tipo como parte del nombre del archivo.

- El nombre se divide en dos partes: un nombre y una extensión, generalmente separadas por un punto.
- De esta forma, el usuario y el sistema determinan el tipo de cada archivo con solo analizar el nombre.
- Sin embargo, estas extensiones no son obligatorias y deben considerarse “sugerencias” dirigidas a las aplicaciones que operan con los archivos.

Tipos de archivo

Otros sistemas utilizan, para identificar el tipo de un archivo, una técnica basada en un *número mágico* almacenado al principio del archivo.

- Sin embargo, no todos los archivos tienen números mágicos, por lo que la funcionalidad del sistema no puede basarse exclusivamente en esta información.

Estructura de los archivos

Los tipos de archivo también pueden usarse para indicar la estructura interna del archivo.

- Por ejemplo, el sistema operativo requiere que los archivos ejecutables tengan una estructura concreta, para poder determinar en qué parte de la memoria cargar el archivo y dónde esta ubicada la primera instrucción.

Estructura de los archivos

Si el sistema operativo soporta múltiples estructuras de archivo, el tamaño resultante del sistema será excesivo.

- Por ello, algunos sistemas imponen (y soportan) un número mínimo de estructuras de archivo.
- Este esquema proporciona una máxima flexibilidad, pero un escaso soporte: cada programa de aplicación debe incluir su propio código para interpretar los archivos de entrada de acuerdo con la estructura apropiada.

Estructura interna de los archivos

Los sistemas de disco suelen tener un tamaño de bloque bien definido, que está determinado por el tamaño del sector.

- Todas las operaciones de E/S de disco se realizan en unidades de un bloque (registro físico) y todos los bloques tienen el mismo tamaño.
- Generalmente el tamaño del registro físico no corresponde exactamente con la longitud deseada de los registros lógicos de un archivo.
- Los registros lógicos pueden incluso variar en longitud dentro de un mismo archivo.
- Una solución común a este problema consiste en *empaquetar* varios registros lógicos dentro de los bloques físicos.
- El empaquetamiento puede ser realizado por la aplicación del usuario o por el sistema operativo.

Estructura interna de los archivos

Puesto que el espacio de disco se asigna siempre en bloques, generalmente se desperdiciará una parte del último bloque de cada archivo.

- Ese desperdicio se conoce con el nombre de *fragmentación interna*.
- Todos los sistemas de archivos sufren de este problema; además, cuanto más grande sea el tamaño del bloque, mayor será el desperdicio.

Métodos de acceso

Puede accederse a la información contenida en los archivos en varios formas distintas.

- Algunos sistemas sólo proporcionan un método de acceso, mientras otros sistemas soportan muchos métodos de acceso y elegir el método adecuado para cada aplicación constituye uno de los principales problemas de diseño.

Acceso secuencial

El método más simple es el *acceso secuencial*. Los datos del archivo se procesan en orden, un registro después de otros.

- Este modo de acceso es el más común.
- Una operación de lectura leerá la siguiente porción del archivo e incrementará automáticamente el puntero al archivo, que controla la ubicación de E/S.
- También, la operación de escritura añadirá datos al final del archivo y avanzará el puntero hasta el final de los datos recién escritos.
- Luego es posible reiniciar el archivo para posicionar el puntero de nuevo al inicio del archivo.

Acceso secuencial

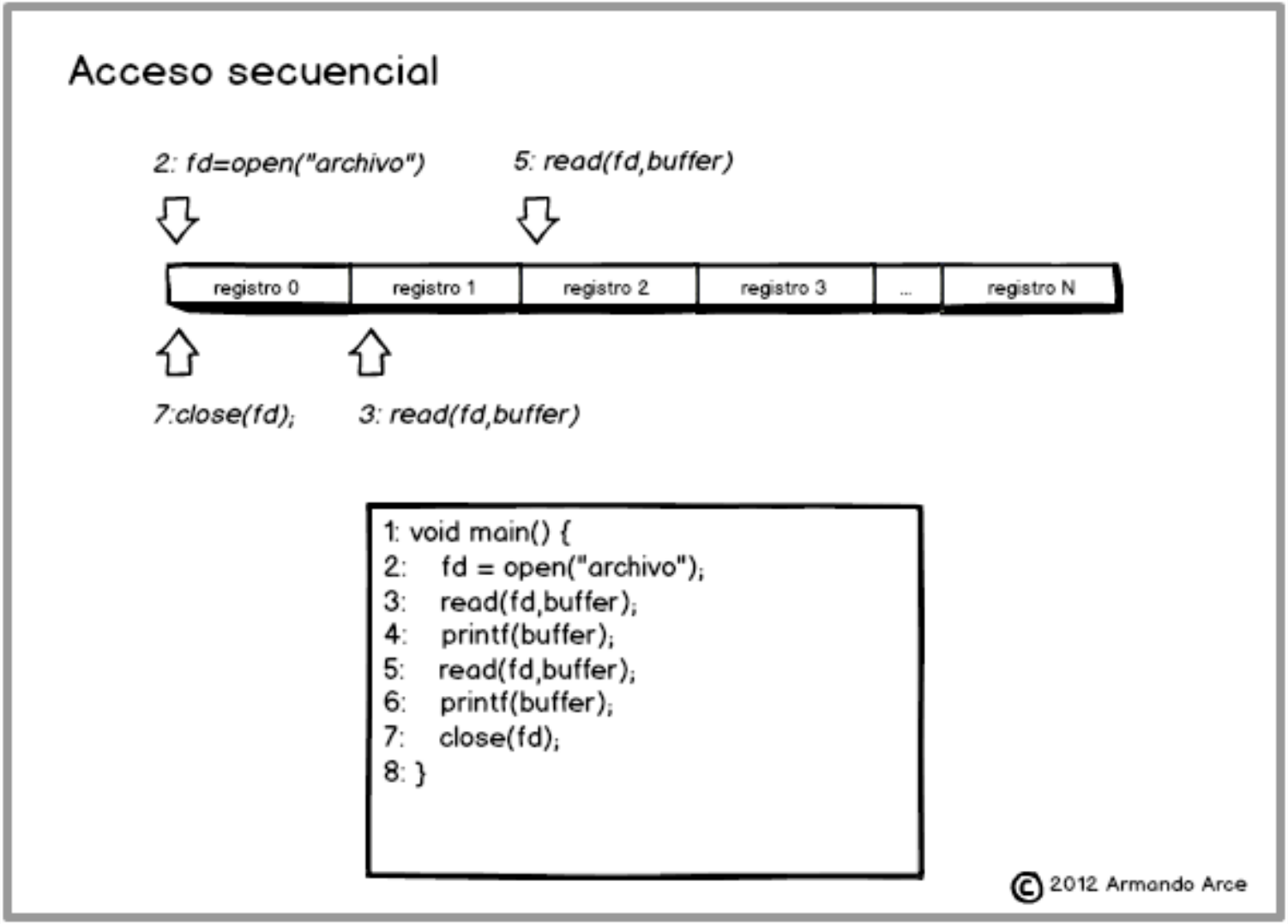

Figure 5:

Acceso directo

Otro método es el *acceso directo* o *acceso relativo*.

- En este modo, el archivo se considera como un secuencia numerada de bloques o registros de tamaño fijo y no existe ninguna restricción en cuanto al orden de lectura o escritura en el archivo.

Acceso directo

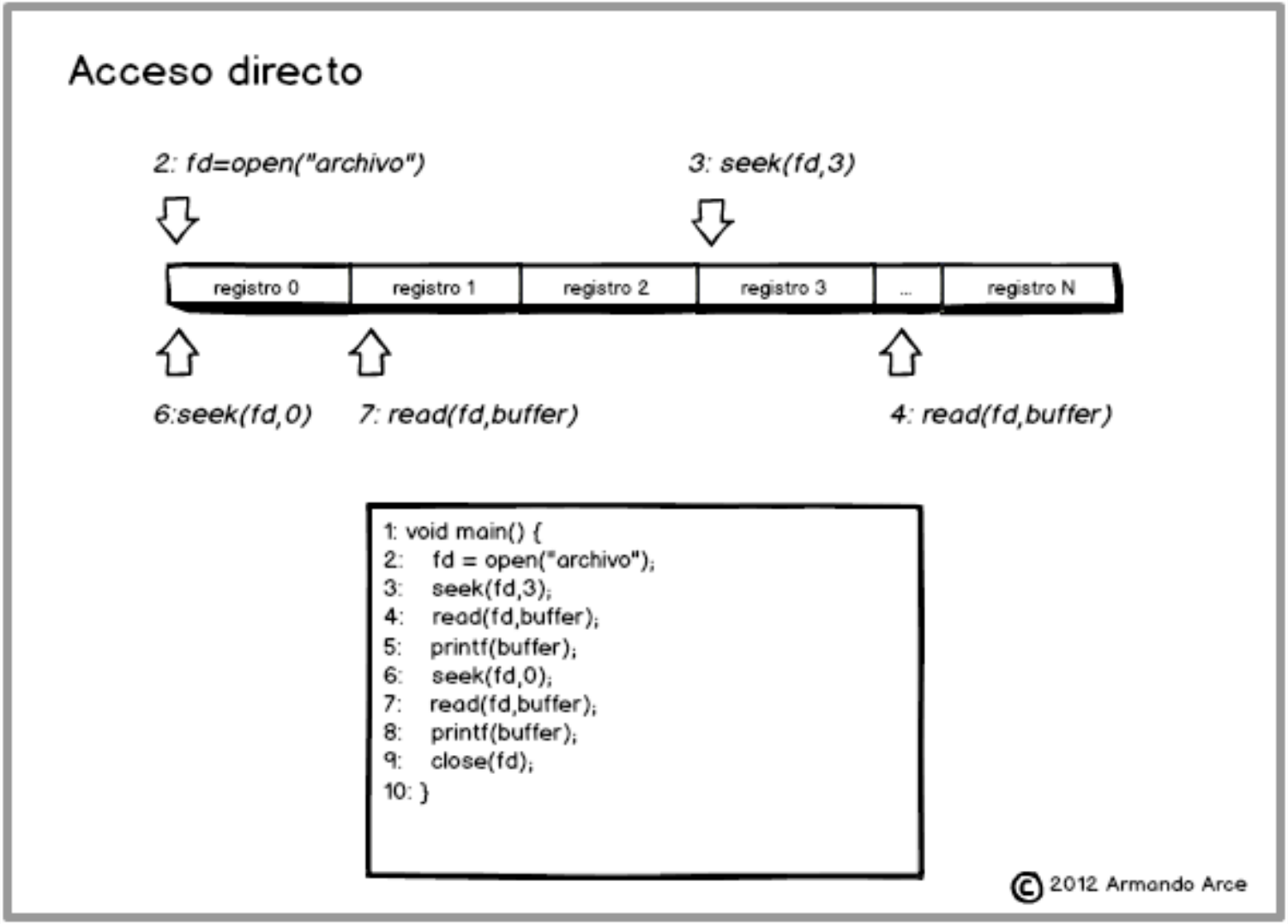

Figure 6:

Acceso directo

En el método de acceso directo, las operaciones de archivo deben modificares para incluir el número de bloque como parámetro.

- Una técnica alternativa consiste en mantener las operaciones de lectura y escritura intactas, e incorporar una operación de *posicionar* (seek) el puntero en cierto bloque.
- El número de bloque proporcionado es normalmente un *número de bloque relativo*, el cual es un índice referido al comienzo del archivo.

Acceso directo

Nótese que para realizar la recuperación directa de un registro lógico en un archivo (y el cálculo del número de bloque relativo) es necesario conocer tanto el tamaño de dicho registro como el tamaño del registro físico (bloque).

- Normalmente el sistema operativo no almacena el tamaño del registro lógico de cada archivo, por lo que resulta más común que este cálculo sea realizado directamente por el programa de usuario.

Otros métodos de acceso

Otro método de acceso es el *índice* que consiste en una estructura o archivo adicional que contiene punteros a los distintos bloques de un archivo.

- Para encontrar un registro dentro del archivo, primero se carga el índice en memoria, se explora y luego se usa el puntero para acceder al archivo directamente y para hallar el registro deseado.
- Generalmente se realiza una búsqueda binaria sobre el índice para acceder rápidamente al puntero adecuado.

Otros métodos de acceso

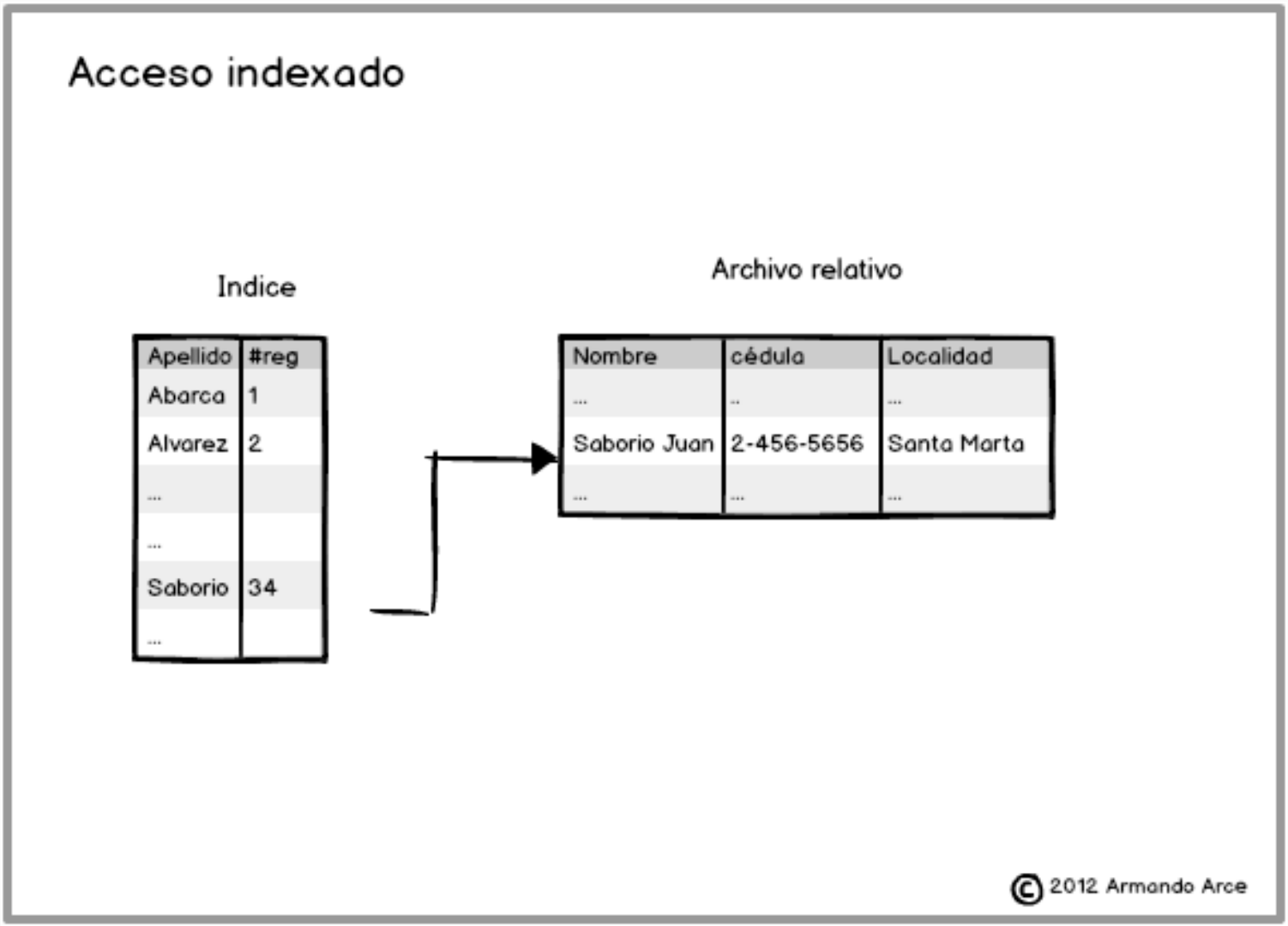

Figure 7:

Otros métodos de acceso

Con los archivos de gran tamaño, el propio archivo de índice puede ser demasiado grande como para almacenarlo en memoria.

- Una solución consiste en crear un índice del archivo de índice (método ISAM de IBM).
- El archivo de índice principal contendría punteros a los archivos de índices secundarios que a su vez apuntarían a los propios elementos de datos.

Estructura de directorios

Para administrar todos los datos de los múltiples archivos de un disco, es necesario organizarlos de alguna forma y esta organización implica el uso de directorios.

Estructura de almacenamiento

En ocasiones es deseable colocar múltiples sistemas de archivos en un mismo disco o utilizar partes de un disco para un sistema de archivos y otras partes para otras cosas (p.ej. el espacio de intercambio).

- Estas partes se conocen con diversos nombres, como *particiones*, franjas o minidiscos.
- Se puede crear un sistema de archivos en cada una de estas partes del disco, pero también se podría combinar las partes para formar estructuras de mayor tamaño, conocidas con el nombre de volúmenes, y también pueden crearse sistemas de archivos en dichos volúmenes.

Estructura de almacenamiento

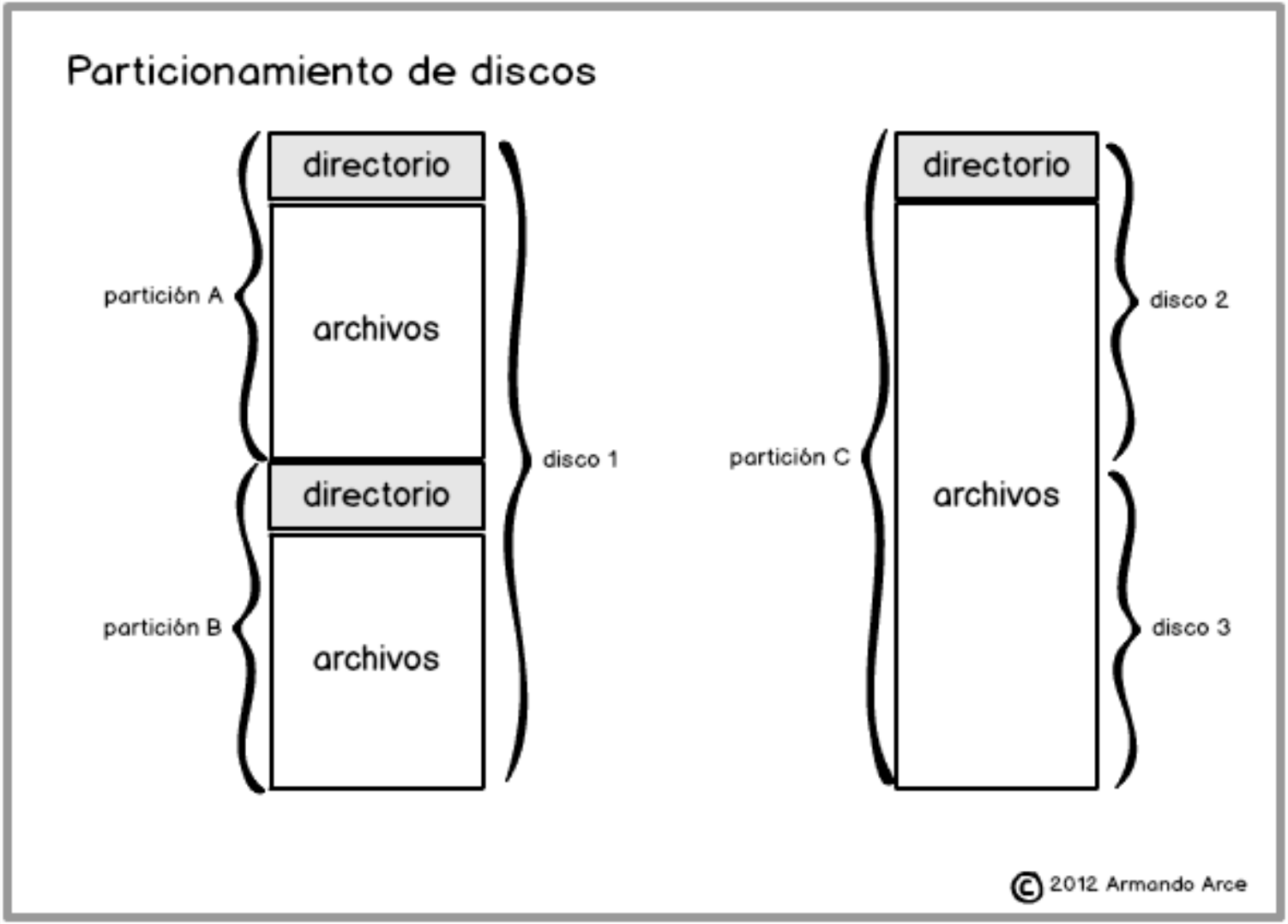

Figure 8:

Estructura de almacenamiento

Cada volumen puede considerarse como si fuera un disco virtual.

- Los volúmenes pueden también almacenar múltiples sistemas operativos, permitiendo que un sistema se inicie en cualquiera de ellos y ejecute dicho sistema operativo.
- Cada volumen que contenga un sistema de archivos debe contener información acerca de los archivos almacenados en el sistema.
- Esta información se almacena como entrada en un *directorio de dispositivo* o *tabla de contenidos de volumen*.
- El directorio de dispositivo almacena información de todos los archivos en el volumen.

Introducción a los directorios

El directorio puede considerarse como una tabla de símbolos que traduce los nombres de archivo a sus correspondientes entradas de directorio.

- Generalmente las operaciones que se llevan a cabo sobre los directorios son: búsqueda de un archivo, crear un archivo (y añadirlo al directorio), borrar un archivo, listar un directorio, renombrar un archivo, y recorrer la jerarquía del sistema de archivos.

Directorio de un único nivel

La estructura de directorio más simple es el directorio de un único nivel.

- Aquí todos los archivos están contenidos en el mismo directorio. Esto resulta fácil de mantener y comprender.
- Sin embargo, todos los archivos deben tener nombres distintos, y es difícil recordar los nombres de todos los archivos.
- También presenta problemas cuando se tienen múltiples usuarios en el sistema pues no existe privacidad en cuanto a los archivos.

Directorio de un único nivel

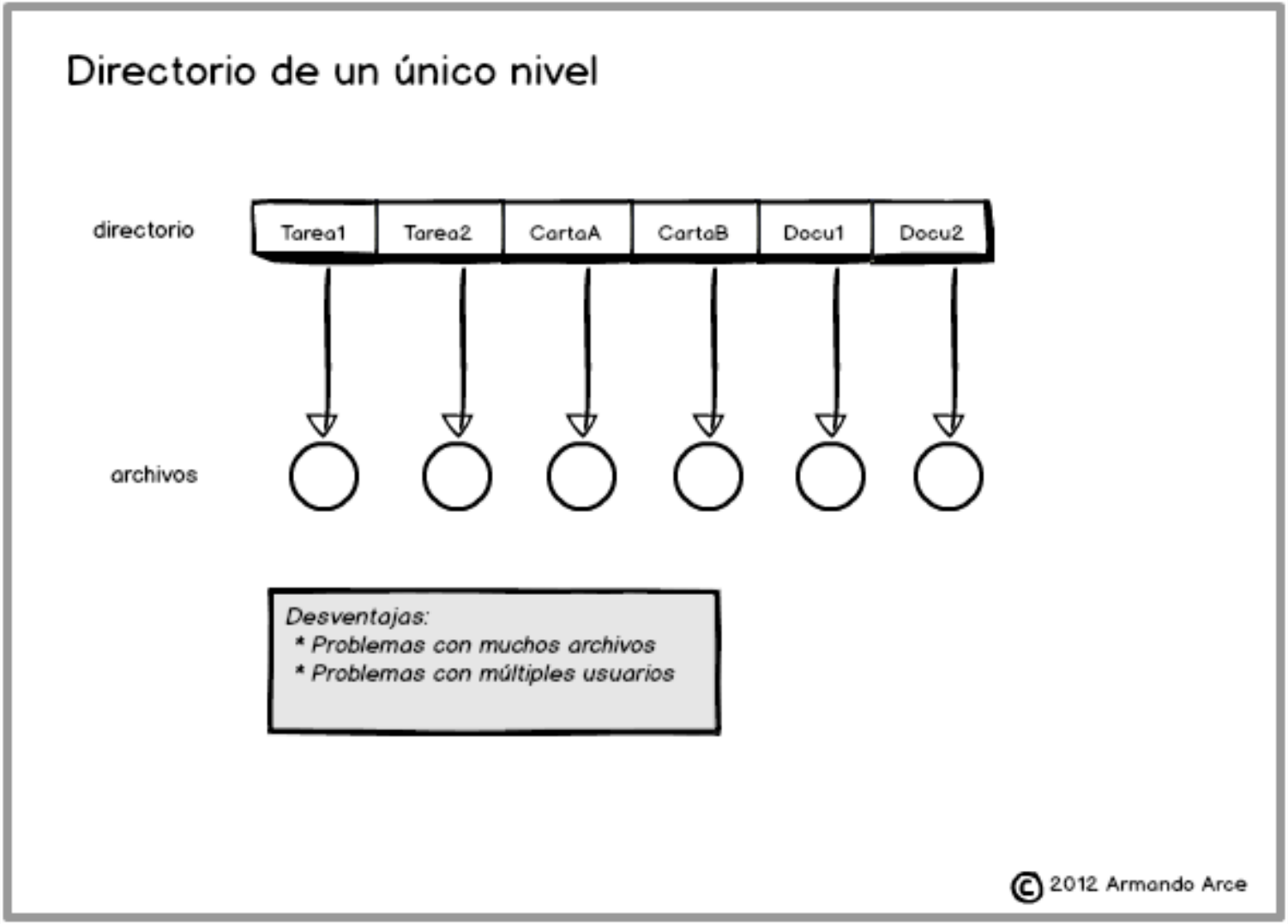

Figure 9:

Directorio de dos niveles

Existe un directorio independiente por cada usuario, generalmente llamado el *directorio de archivos de usuario* (UFD, user file directory).

- Cuando un usuario hace referencia a un archivo concreto, sólo se explora su propio UFD.
- Por tanto, este esquema soluciona el problema de colisión de nombres entre usuarios.
- Sin embargo, no permite a los usuarios cooperar en una cierta tarea y poder acceder a los archivos de otros usuarios.

Directorio de dos niveles

Aún peor es el problema cuando se intenta acceder a los archivos del sistema.

- Una posible solución sería copiar todos los archivos del sistema al directorio del usuario pero no sería muy práctico, por ello generalmente se prefiere complicar el procedimiento de búsqueda un poco y añadir la exploración de un directorio especial del sistema.

Directorio de dos niveles

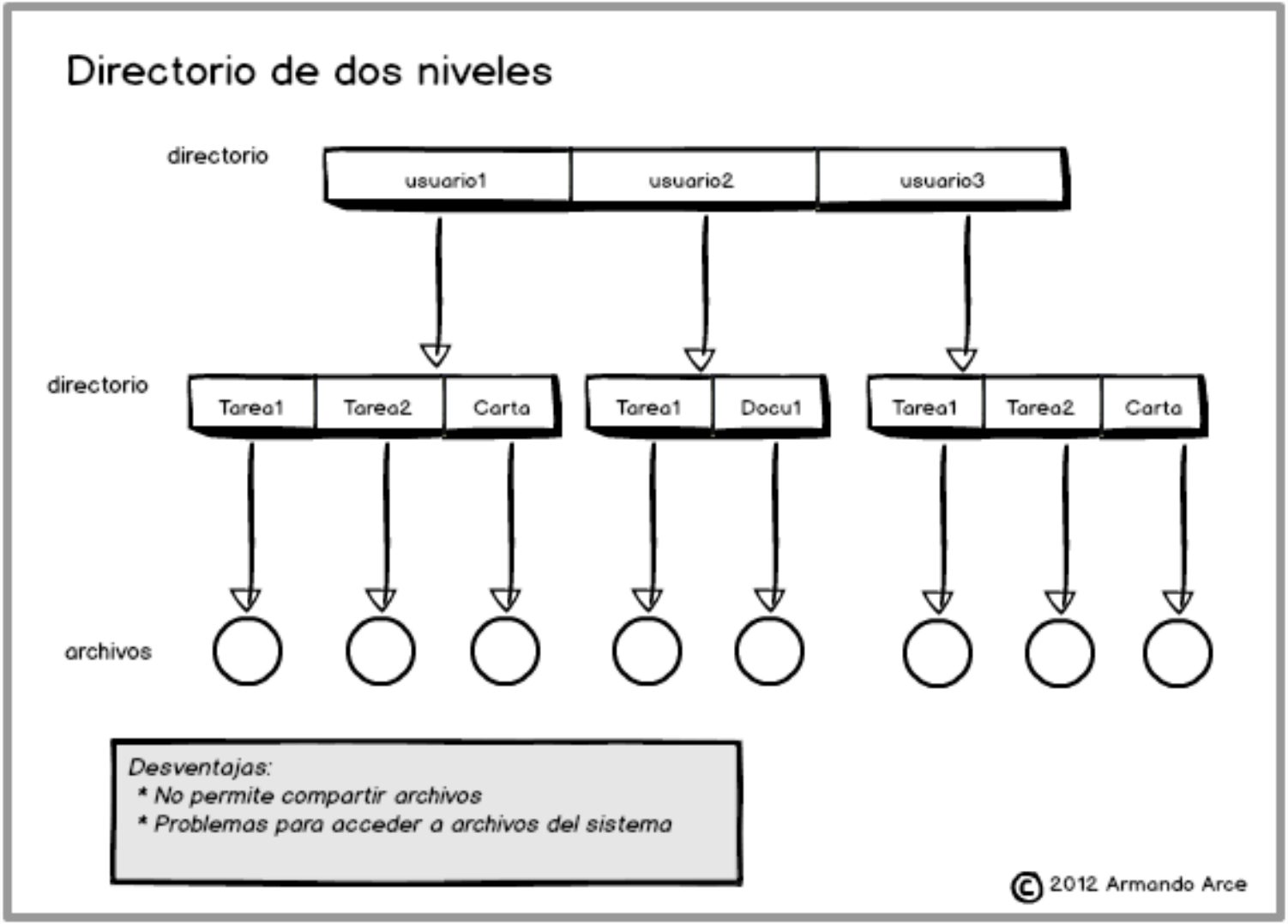

Figure 10:

Directorios con estructura de árbol

La estructura de directorios se amplia para crear un árbol de altura arbitraria.

- Los usuarios pueden crear sus propios subdirectorios y organizar sus archivos correspondientemente.
- Con un sistema de directorios con estructura de árbol, se puede permitir que los usuarios accedan a los archivos de otros usuarios, además de acceder a los suyos propios.

Directorios con estructura de árbol

Sin embargo, una *ruta a un archivo* (la secuencia de directorios a los que se debe ingresar para llegar al archivo) en un directorio con estructura de árbol puede ser más larga que las rutas típicas de los directorios en dos niveles.

- Por ello, se utiliza el concepto de *directorio actual* y se definen rutas absolutas y relativas.
- Un nombre de *ruta absoluta* comienza en la raíz y sigue una ruta descendente hasta el archivo especificado.
- Un *ruta relativa* define la ruta a partir del directorio actual.
- Además, el sistema también provee una operación (cd, change directory) para realizar el cambio del directorio actual.

Directorios con estructura de árbol

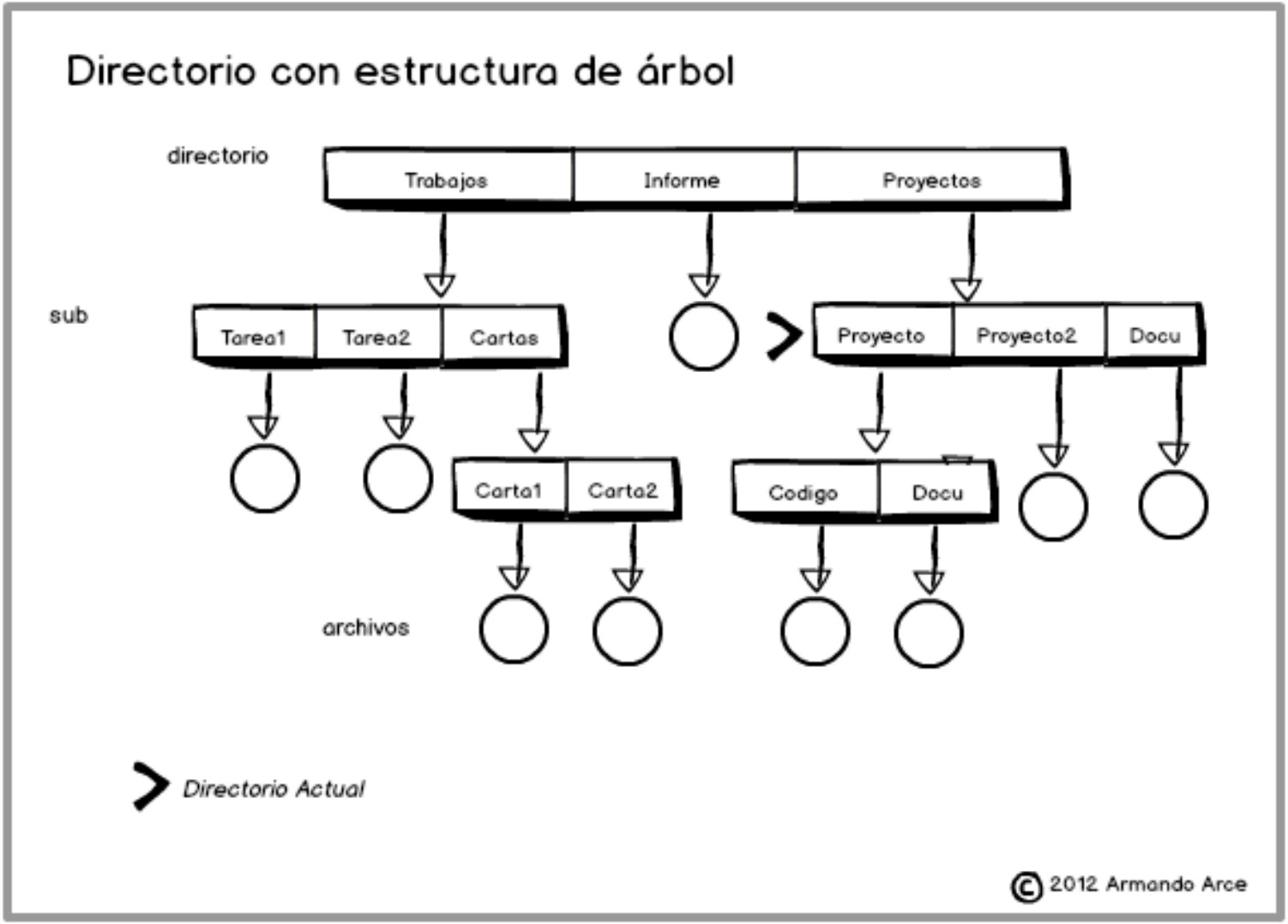

Figure 11:

Directorios con estructura de árbol

Una aspecto importante es qué hacer si se borra un directorio.

- Una opción es que el sistema no permita borrar un directorio a menos que esté vacío, otra opción sería borrar todos los archivos y subdirectorios del directorio inicial.
- La primera opción puede requerir mucho trabajo adicional por parte del usuario y, aunque la segunda opción es la más cómoda, también es la más peligrosa porque puede eliminarse una estructura de directorios completa con un único comando.

Directorios en un grafo acíclico

Un grafo acíclico permite que los directorios compartan subdirectorios de archivos.

- El mismo archivo o subdirectorio puede estar en dos directorios diferentes simultáneamente.
- El grado cíclico es una generalización natural del esquema de directorio con estructura de árbol.

Directorios en un grafo acíclico

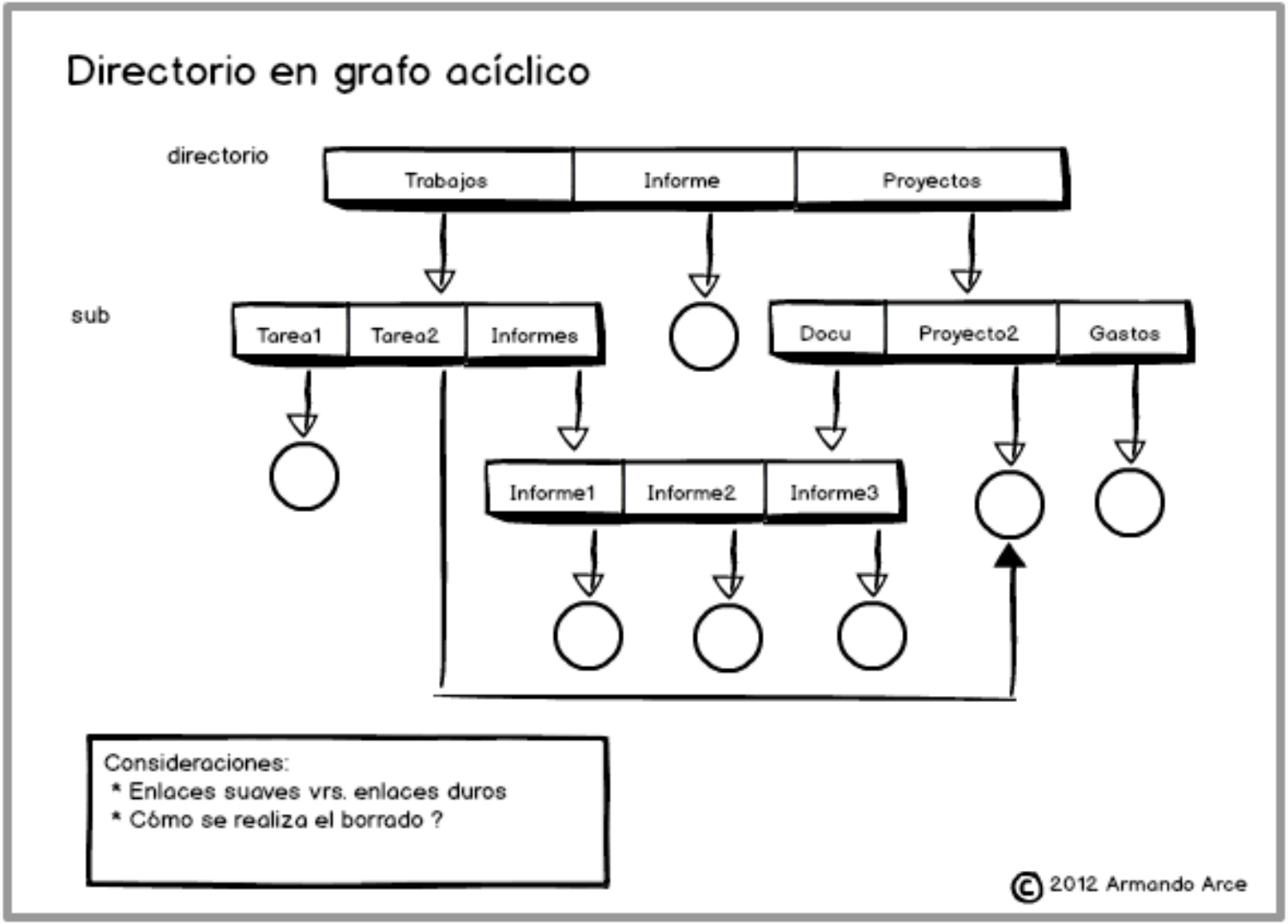

Figure 12:

Directorios en un grafo acíclico

Es importante observar que un archivo (o directorio) compartido no es lo mismo que dos copias del archivo.

- Si una persona modifica un archivo compartido que se encuentra en su propio directorio, el cambio será visible para otras personas que mantengan en sus propios directorios el archivo compartido.
- Aquí sólo existe un archivo real.

Directorios en un grafo acíclico

Los archivos y subdirectorios compartidos pueden implementarse de diversas formas. Una manera bastante común consiste en crear una nueva entrada de directorio denominada enlace suave (soft link).

- Un enlace suave es, en la práctica, un puntero a otro archivo o subdirectorio.
- Este se puede implementar como un nombre de ruta absoluta o relativa. Los enlaces generalmente se identifican por la extensión de su nombre de archivo.
- Un problema que presenta este método es que si el archivo original cambia de ubicación o es borrado, todos los enlaces a dicho archivo quedarán inconsistentes.
- Una opción sería identificar todos los enlaces al archivo que cambia de ubicación y actualizarlos de forma adecuada, pero esta solución bastante costosa en la práctica.

Directorios en un grafo acíclico

Otra técnica común para implementar archivos compartidos consiste en duplicar toda la información acerca del archivo en todos los directorios que compartan dicho archivo (enlaces fuertes).

- Así, ambas entradas serán idénticas y el original y la copia serán indistinguibles.
- El principal problema con las entradas de directorio duplicadas es el de mantener la coherencia cuando se modifica un archivo (no cuando cambia de directorio).
- Por ejemplo, los cambios de nombre, tamaño, y derechos de acceso deben aplicarse a todas las entradas.

Directorios en un grafo acíclico

Este tipo de estructura es más flexible, pero también más compleja. Los archivos podrán tener ahora múltiples nombres de ruta absoluta lo que puede provocar que se recorran múltiples veces las estructuras compartidas en búsquedas recursivas.

- Otro problema es lo que se refiere al borrado pues no es fácil determinar cuándo eliminar del disco un archivo que está siendo compartido.

Directorios en un grafo acíclico

- Para el caso de enlaces suaves, el borrado de un enlace no provoca ningún problema pero el borrado del archivo original deja todos los enlaces inconsistentes.
- Para el caso de enlaces fuertes es recomendable manejar un contador de enlaces que lleve la cuenta de la cantidad de referencias a un archivo determinado, en cada borrado se disminuirá la cuenta en uno, al llegar a cero se liberará el espacio asignado a dicho archivo.

Directorios en un grafo general

En este caso los recorridos se pueden ciclar (retornar al mismo subdirectorio que ya se había visitado).

- Si se permiten que existan ciclos en el directorio, se necesita de una forma de evitar tener que buscar en cualquier directorio dos o más veces, principalmente por razones de rendimiento.
- Una solución consiste en limitar arbitrariamente el número de directorios a los que accederá durante una búsqueda.
- Otro método más costoso es aplicar un algoritmo de reconocimiento de ciclos en grafos.

Directorios en un grafo general

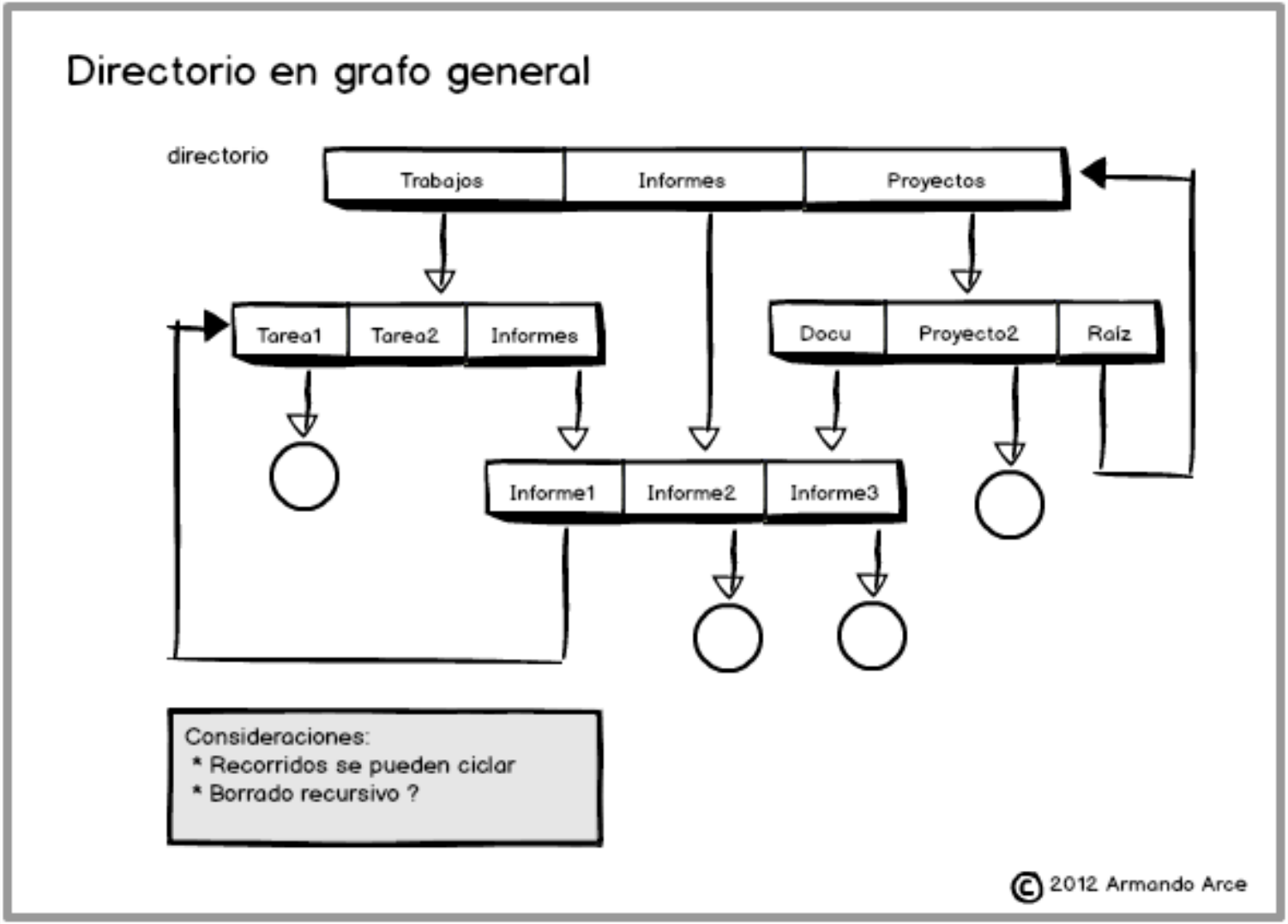

Figure 13:

Directorios en un grafo general

Un problema similar existe cuando se trata de determinar si un archivo puede ser borrado.

- Con las estructuras de grado cíclico, un valor de 0 en el contador de referencias significará que ya no hay más referencias al archivo o directorio, y que ese archivo puede ser borrado.
- Sin embargo, si existen ciclos ese contador puede no ser 0 aún cuando ya no existan referencias a ese directorio, pues puede existir una auto-referencia en la estructura de directorios.
- En este caso generalmente se utiliza un esquema de *recolección de memoria* para determinar cuándo se ha borrado la última referencia.

Directorios en un grafo general

La recolección de memoria implica recorrer todo el sistema de archivos, marcando todos aquellos elementos a los que pueda acceder.

- Después, una segunda pasada recopila todo aquello que no está marcado, incluyéndolo en una lista de espacio libre.
- Sin embargo, este procedimiento consume muchísimo tiempo y en pocas ocasiones se suele utilizar (p.ej. una vez después de múltiples encendidos de una máquina).

Montaje de sistemas de archivos

Un sistema de archivos puede estar formada por múltiples volúmenes que deben poder *montarse* para estar disponibles.

- El proceso de montaje es bastante simple. Se le proporciona al sistema operativo el nombre del dispositivo y el *punto de montaje* que es la ubicación dentro de la estructura de archivos a la que hay que conectar el sistema de archivos que se está montando.
- Normalmente, el punto de montaje es un directorio vacío.
- A continuación, el sistema verifica que el dispositivo contiene un sistema de archivos válido (el directorio tiene el formato esperado).
- Finalmente, el sistema registra en su estructura de directorios que ya un sistema de archivo montado en el punto de montaje especificado.

Montaje de sistemas de archivos

Los sistemas imponen una cierta semántica para clarificar la funcionalidad.

- Por ejemplo, un determinado sistema podría prohibir que se realizara un montaje en un directorio que contuviera archivos, o bien podría hacer que el sistema de archivos montado estuviera disponible en dichos directorios y ocultar los archivos existentes hasta que el sistema de archivos se desmontara.
- En otro ejemplo, un determinado sistema podría permitir que se montara repetidamente el mismo sistema de archivos, en diferentes puntos de montaje; o por el contrario, podría permitir un único montaje por cada sistema de archivos.

Referencia bibliográfica

A.SILBERSCHATZ, P. GALVIN, y G. GAGNE, Operating Systems Concepts, Cap.10, 9a Edición, John Wiley, 2013.