

Este documento contiene las pizarras y ejemplos de código que se utilizaron en las clases de Programación I del grupo G1M3, durante el mes de diciembre de 2025, dedicadas a los arrays compactados y los arrays con huecos.

### 1 . Clase 28 noviembre 2025: Arrays compactados

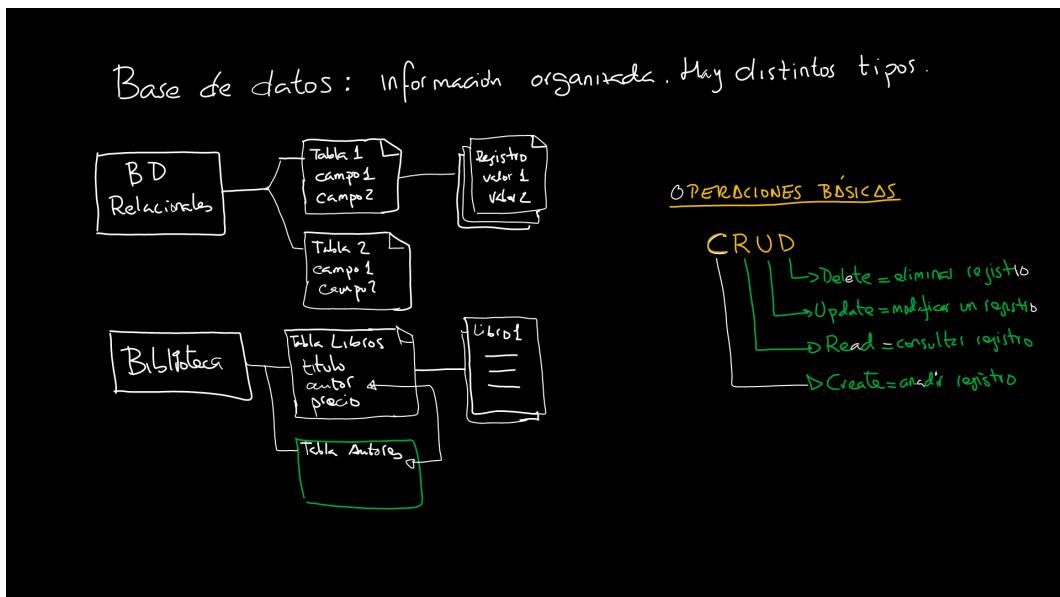


Figura 1: Bases de datos: concepto y operaciones básicas.

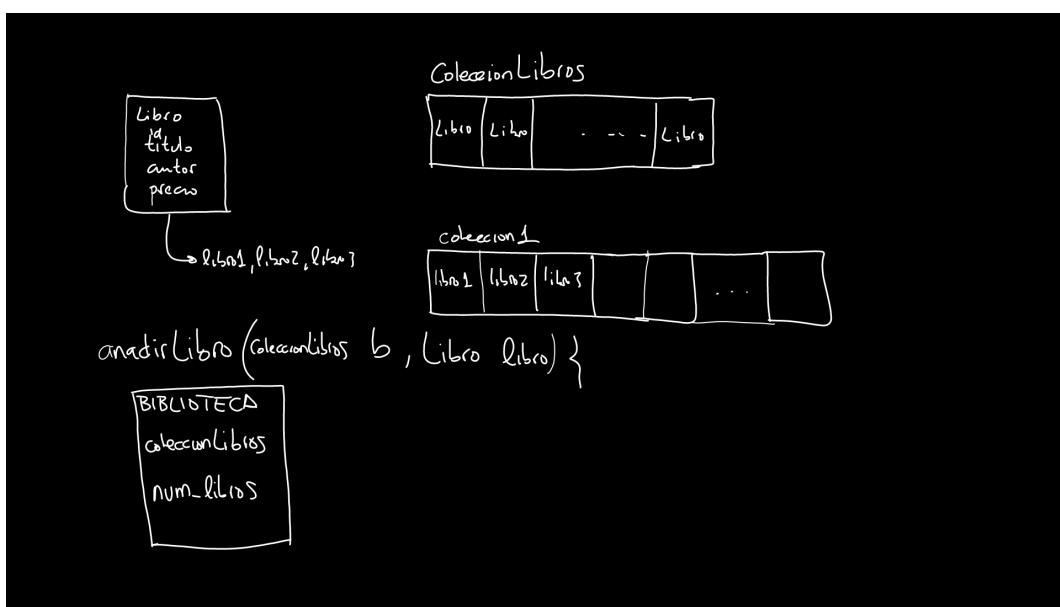


Figura 2: Técnica del array compactado para gestionar bases de datos.

2 . Clase 3 diciembre 2025: Arrays compactados y arrays con huecos

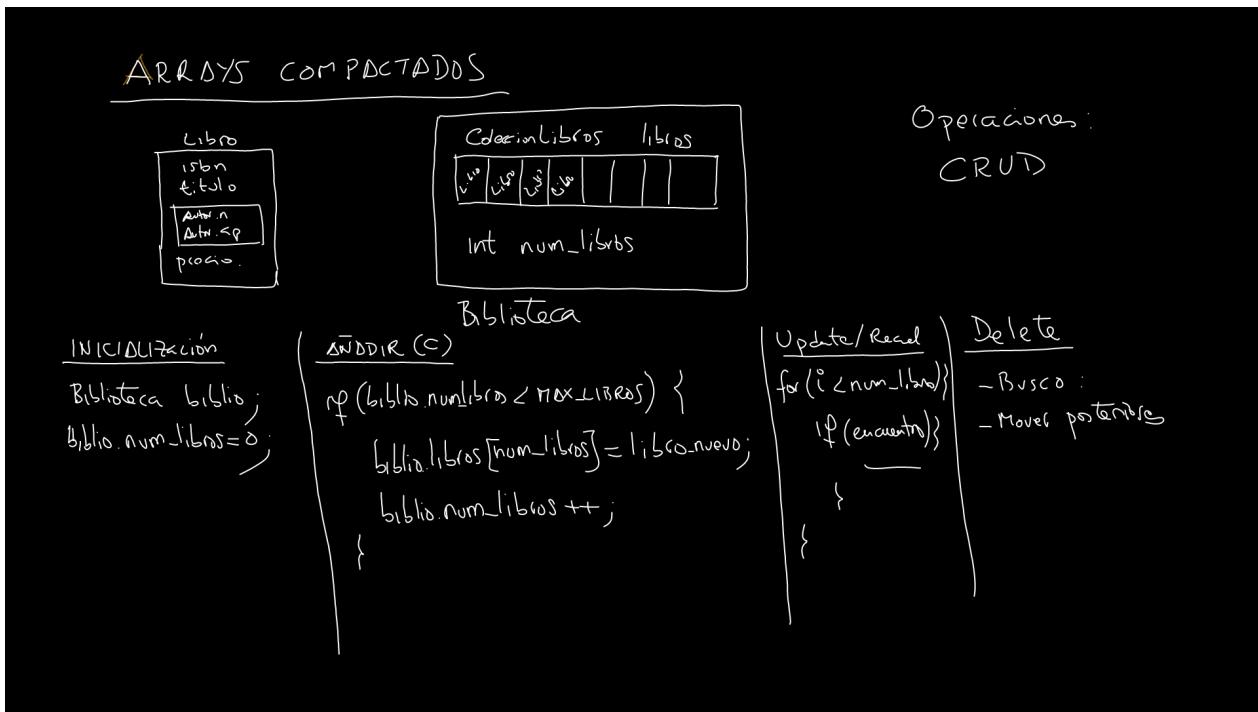


Figura 3: Ejemplo: gestionar una Base de Datos de Libros mediante la técnica del array compactado.

El Ejemplo 1 muestra la gestión de una base de datos de libros mediante la técnica del array compactado. Se codifican dos de las operaciones CRUD: añadir un registro a la base de datos (*CREATE*) y borrar un registro de la base de datos (*DELETE*).

## Ejemplo 1 Ejemplo de gestión de B.D con array compactado

```
#include <stdbool.h>
#include <stdio.h>
#include <string.h>

// Definir constantes
#define MAX_LIBROS 100

// Tipos de datos
typedef struct {
    int id;
    char titulo[80];
    char autor[40];
    double precio;
} Libro;

typedef Libro ColeccionLibros[MAX_LIBROS];

typedef struct {
    ColeccionLibros libros;
    int num_libros;
} Biblioteca;

// Prototipos de funciones
int anadirLibro(Biblioteca* bib, Libro libro);
int borrarLibro(Biblioteca* bib, char titulo[]);
```

```

// Programa principal
int main() {
    // Inicializar biblioteca
    Biblioteca biblio;
    biblio.num_libros = 0;

    // Añadir tres libros
    Libro libro1 = {1, "La colmena", "Camilo J. Cela", 3.14};
    anadirLibro(&biblio, libro1);
    Libro libro2 = {2, "El Quijote", "Miguel de Cervantes"};
    anadirLibro(&biblio, libro2);
    Libro libro3 = {3, "El perro de Baskerville", "Arthur C.D"};
    anadirLibro(&biblio, libro3);

    // Si modificamos la variable libro3, después de haber llamado
    // al método añadir, la biblioteca no se modifica. Cuando llamamos al
    // método anadirLibro(), la función recibe una copia de libro3. La
    // variable libro3 es independiente de la copia que hay almacenada
    // en la biblioteca.
    strcpy(libro3.titulo, "El sabueso de Baskerville");

    // Listar los títulos de los libros de la biblioteca
    printf("-----LISTADO DE LIBROS -----\\n");
    printf("Número de libros en la biblioteca: %d\\n", biblio.num_libros);
    for (int i = 0; i < biblio.num_libros; i++) {
        printf("%s\\n", biblio.libros[i].titulo);
    }
    printf("-----\\n\\n");

    // Tras codificar la función borrarLibro(), conviene hacer pruebas
    // del funcionamiento correcto de la función.
    int resultado;
    // printf("Borrando un libro en posición intermedia...\\n");
    // resultado = borrarLibro(&biblio, "El Quijote");
    // printf("Resultado: %d\\n\\n", resultado);
    // printf("Borrando el primero de la lista...\\n");
    // resultado = borrarLibro(&biblio, "La colmena");
    // printf("Resultado: %d\\n\\n", resultado);
    // printf("Borrando el último de la lista...\\n");
    // resultado = borrarLibro(&biblio, "El perro de Baskerville");
    // printf("Resultado: %d\\n\\n", resultado);
    printf("Intentar borrar un libro que no existe...\\n");
    resultado = borrarLibro(&biblio, "El sabueso de Baskerville");
    printf("Resultado: %d\\n\\n", resultado);

    printf("-----LISTADO DE LIBROS -----\\n");
    printf("Listado tras pruebas de borrarLibro():\\n");
    printf("-----\\n");
    printf("Número de libros en la biblioteca: %d\\n", biblio.num_libros);
    for (int i = 0; i < biblio.num_libros; i++) {
        printf("%s\\n", biblio.libros[i].titulo);
    }
    printf("-----\\n");

    return 0;
}

```

```

// Código de funciones

// anadirLibro
// Añade un libro a la base de datos
// Primero comprueba si hay sitio. Si lo hay, añade
// el nuevo libro a continuación del último
// existente en la B.D. y actualiza el contador de libros.
// Devuelve: 1, si se añadió el libro o 0, si no se pudo añadir.
int anadirLibro(Biblioteca* bib, Libro libro) {
    int result = 0;
    if (bib->num_libros < MAX_LIBROS) {
        bib->libros[bib->num_libros] = libro;
        bib->num_libros += 1;
        result = 1;
    }
    return result;
}

// borrarLibro
// Borra un libro de la base de datos, desplazando los libros
// posteriores al que se borra, una posición hacia la izquierda
// Devuelve: 1, si se borró el libro o 0, si no se pudo borrar.
int borrarLibro(Biblioteca* bib, char titulo[]) {
    int result = 0;
    int n = bib->num_libros;
    bool seguir = true;
    int i = 0;
    while (seguir == true && i < bib->num_libros) {
        if (strcmp(bib->libros[i].titulo, titulo) == 0) {
            // Mover los libros posteriores al i una pos a la izquierda
            for (int j = i + 1; j < bib->num_libros; j++) {
                bib->libros[j - 1] = bib->libros[j];
            }
            // Actualizar contador de libros
            bib->num_libros -= 1;
            result = 1;
            seguir = false;
        } else {
            i = i + 1;
        }
    }
    return result;
}

```

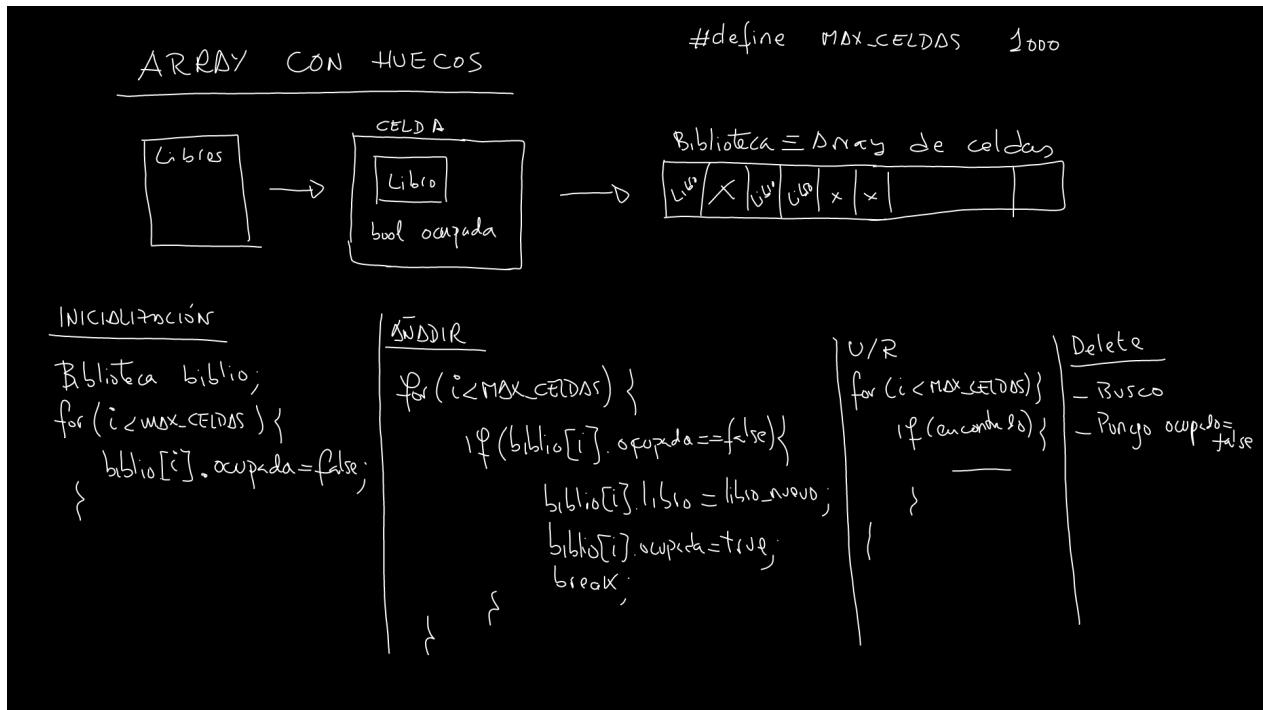


Figura 4: Ejemplo: gestionar una Base de Datos de Libros mediante la técnica del array con huecos.

El Ejemplo 2 muestra la gestión de una colección de libros usando la técnica del array con huecos. Se codifican las operaciones CRUD Create y Delete.

#### Ejemplo 2 Gestión de una B.D. de libros mediante array con huecos

```

#include <stdbool.h>
#include <stdio.h>
#include <string.h>

// Constantes
#define MAX_CELDAS 100

// Tipos de datos
typedef struct {
    int id;
    char titulo[80];
    char autor[40];
    double precio;
} Libro;

typedef struct {
    Libro libro;
    bool ocupada;
} Celda;

typedef Celda Biblioteca[MAX_CELDAS];

// Prototipos
int listarLibros(const Biblioteca bib);
int anadirLibro(Biblioteca biblio, Libro libro_nuevo);
int borrarLibro(Biblioteca bib, char tit[]);

```

```

// Programa principal
int main() {
    // Inicializar BD
    Biblioteca biblio;
    for (int i = 0; i < MAX_CELDAS; i++) {
        biblio[i].ocupada = false;
    }

    // Añadir tres libros
    Libro libro1 = {1, "La colmena", "Camilo J. Cela", 3.14};
    anadirLibro(biblio, libro1);
    Libro libro2 = {2, "El Quijote", "Miguel de Cervantes"};
    anadirLibro(biblio, libro2);
    Libro libro3 = {3, "El perro de Baskerville", "Arthur C.D"};
    anadirLibro(biblio, libro3);

    listarLibros(biblio);

    // Borrar un título
    char tit[] = "El Quijote";
    int result = borrarLibro(biblio, tit);
    if(result == 1) {
        printf("Se borró el libro %s de la B.D.\n", tit);
    } else {
        printf("No se pudo borrar el libro %s de la B.D.", tit);
    }

    listarLibros(biblio);

    return 0;
}

// Código de funciones

// listarLibros
// Lista los libros de la base de datos
// Devuelve el número de libros existente en la B.D.
int listarLibros(const Biblioteca bib) {
    int contador = 0;
    printf("\n----- LISTADO DE LIBROS -----");
    for (int i = 0; i < MAX_CELDAS; i++) {
        if (bib[i].ocupada == true) {
            Libro libro = bib[i].libro;
            printf("%s %s\n", libro.titulo, libro.autor);
            // printf("%s %s\n", bib[i].libro.titulo, bib[i].libro.autor);
            contador++;
        }
    }
    printf("-----\n\n");
    return contador;
}

```

```

// anadirLibro
// Añade un libro a la base de datos.
// Devuelve 1, si se añade el libro y 0, en caso contrario
int anadirLibro(Biblioteca bib, Libro libro_nuevo) {
    int result = 0;
    bool seguir = true;
    int indice = 0;
    while (seguir == true && indice < MAX_CELDAS) {
        if (bib[indice].ocupada == false) { // if(!bib[indice].ocupada) {}
            bib[indice].libro = libro_nuevo;
            bib[indice].ocupada = true;
            result = 1;
            seguir = false;
        } else {
            indice++;
        }
    }
    return result;
}

// borrarLibro
// Borra un libro de la base de datos.
// Devuelve 1, si se borra, 0 en caso contrario
int borrarLibro(Biblioteca bib, char tit[]) {
    int result = 0;
    bool seguir = true;
    int indice = 0;
    while (seguir == true && indice < MAX_CELDAS) {
        if (bib[indice].ocupada == true) {
            if (strcmp(tit, bib[indice].libro.titulo) == 0) {
                bib[indice].ocupada = false;
                result = 1;
                seguir = false;
            }
        }
        indice++;
    }
    return result;
}

```