



Universidad Autónoma de Baja California

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA



Reporte de actividad

Lenguaje C (531).

Práctica No. 13:

Implementación de estructuras de datos con arreglos.

Alumno

Joshua Osorio Osorio - 1293271

Docente

Fernando Elihonai Saucedo Lares

5 de mayo de 2025

ÍNDICE

Objetivo	2
Introducción	2
Lista de materiales	2
Desarrollo	2
Código <main.c>:	2
Conclusiones	6
Bibliografía	6

Objetivo

Evaluar el manejo de estructuras de datos y el uso de funciones con parámetros para organizar el código de manera eficiente.

Introducción

El programa utiliza estructuras de datos para modelar jugadores y cartas, demostrando:

Uso de estructuras (struct): Para organizar datos relacionados (ej: JUGADOR con nombre, cartas y contador de victorias).

Funciones modularizadas: Cada acción del juego (asignar cartas, jugar rondas) se encapsula en funciones.

Aleatoriedad: Generación de valores aleatorios para simular el reparto de cartas.

Lista de materiales

- Computadora
- Compilador GCC (MinGW).
- Editor de código (bloc de notas =().

Desarrollo

Descripción

Crear un programa que:

- Asigna 5 cartas a cada jugador(3) con un valor aleatorio del 0 al 9 y un nombre específico.
- Juega rondas donde cada jugador revela una carta y gana el que tenga el valor mayor.
- Al final, determina cuál jugador ganó más manos.

Cada carta tendrá un valor aleatorio entre 0 y 9.

Detalles del programa

-Estructura:

- Jugador
- Cartas (5)
- Manos Ganadas

-Funciones:

- AsignarCartas: Asigna valores a las cartas de cada jugador de manera aleatoria.
- JugarRondas: Realiza una comparación de cartas por rondas, actualizando las manos ganadas de cada jugador.
- DeterminarGanador: Determina quién ganó el juego según la cantidad de manos ganadas.

Función Principal (main):

- Inicializa el juego, llamando a las funciones asignarCartas, jugarRondas y determinarGanador para ejecutar el juego completo.

Código <main.c>:

C/C++

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>

#define NUM_JUGADORES 3
#define NUM_CARTAS 5
```

```

typedef struct
{
    char *nombre;
    int valor;
} CARTA;

typedef struct
{
    char nombre[20];
    CARTA cartas[NUM_CARTAS];
    int manosGanadas;
} JUGADOR;

int Aleatorio(int min, int max);
void AsignarCartas(JUGADOR *jugadores, int numJugadores);
void JugarRondas(JUGADOR *jugadores, int numJugadores);
void DeterminarGanador(JUGADOR *jugadores, int numJugadores);

int main(void)
{
    srand(time(NULL)); /* Inicializar semilla para números aleatorios */
    JUGADOR jugadores[NUM_JUGADORES];
    AsignarCartas(jugadores, NUM_JUGADORES);
    JugarRondas(jugadores, NUM_JUGADORES);
    DeterminarGanador(jugadores, NUM_JUGADORES);
    return 0;
}

int Aleatorio(int min, int max)
{
    return rand() % (max - min + 1) + min;
}

void AsignarCartas(JUGADOR *jugadores, int numJugadores)
{
    /* valor respecto a indice: 0      1      2      3      4      5
6      7      8      9 */
    char *nombresCartas[10] = {"Dos", "Tres", "Cuatro", "Cinco", "Seis",
    "Siete", "Sota", "Caballo", "Rey", "As"};

    for (int i = 0; i < numJugadores; i++)
    {

```

```

        snprintf(jugadores[i].nombre, sizeof(jugadores[i].nombre), "Jugador
%d", i + 1);
        /* snprintf para formatear y guardar cadenas */
        jugadores[i].manosGanadas = 0;

        /* Asignar de una nombre y valor a las n cartas. */
        for (int carta = 0; carta < NUM_CARTAS; carta++)
        {
            int ran = Aleatorio(0, 9);
            jugadores[i].cartas[carta].valor = ran;
            jugadores[i].cartas[carta].nombre = nombresCartas[ran];
        }
    }
}

void JugarRondas(JUGADOR *jugadores, int numJugadores)
{
    for (int ronda = 0; ronda < NUM_CARTAS; ronda++)
    {
        printf("\n----- Ronda %d ----- \n", ronda + 1);

        int max_valor = -1;
        int jugador_ganador = -1;

        for (int jugador = 0; jugador < numJugadores; jugador++)
        {
            printf("%s juega: %s (Valor: %d)\n",
                jugadores[jugador].nombre,
                jugadores[jugador].cartas[ronda].nombre,
                jugadores[jugador].cartas[ronda].valor);

            if (jugadores[jugador].cartas[ronda].valor > max_valor)
            {
                max_valor = jugadores[jugador].cartas[ronda].valor;
                jugador_ganador = jugador;
            }
        }

        if (jugador_ganador != -1)
        {
            jugadores[jugador_ganador].manosGanadas++;
            printf("Gana %s!\n", jugadores[jugador_ganador].nombre);
        }
    }
}

```

```

}

void DeterminarGanador(JUGADOR *jugadores, int numJugadores)
{
    printf("\n--- Resultados Finales ---\n");

    int max_manos = -1;
    int ganador = -1;

    for (int jugador = 0; jugador < numJugadores; jugador++)
    {
        printf("%s: %d manos ganadas\n", jugadores[jugador].nombre,
            jugadores[jugador].manosGanadas);

        if (jugadores[jugador].manosGanadas > max_manos)
        {
            max_manos = jugadores[jugador].manosGanadas;
            ganador = jugador;
        }
    }

    if (ganador != -1)
    {
        printf("\n¡El ganador es %s con %d manos!\n",
            jugadores[ganador].nombre, jugadores[ganador].manosGanadas);
    }
    else
    {
        printf("\n¡Empate!\n");
    }
}

```

Conclusiones

En este proyecto se logró implementar un juego funcional utilizando estructuras y funciones, lo que permitió una organización eficiente y modular del código. La incorporación de la función `rand()` junto con arreglos de cadenas de texto (strings) facilitó la simulación realista del reparto de cartas, asegurando además la correcta asignación de nombres sin provocar desbordamientos de memoria. El uso de apuntadores resultó clave para optimizar el manejo de las estructuras, ya que al pasar parámetros por referencia se mejoró el rendimiento del programa y se logró una gestión más eficiente de la memoria.

Bibliografía

[1] "IBM i," 08-Apr-2025. [Online]. Available:

<https://www.ibm.com/docs/es/i/7.5.0?topic=functions-sprintf-print-formatted-data-buffer>.

[2] "Asignar valor a cadena de caracteres en estructura," Stack Overflow En Español.

[Online]. Available:

<https://es.stackoverflow.com/questions/570958/asignar-valor-a-cadena-de-caracteres-en-estructura>.