

Aspectos Técnicos

Toda la información descrita en el siguiente documento es de carácter obligatorio incluirlo en la tarea, por favor incluirlos.

Main

Structs definidos en main.h

```
1 typedef struct {
2     Tablero* tablero; // Puntero al struct Tablero
3     Ingrediente** inventario;
4     Pedido* pedidos;
5     int turnos_restantes;
6     int dificultad;
7 } Juego;
8 typedef struct {
9     int x;           // Posicion X en el tablero
10    int y;           // Posicion Y en el tablero
11    int en_llamas;   // 1 si el jugador esta en llamas (por
                     // atravesar estacion incendiada)
12 } Jugador;
```

Tablero

- Estructura: Puntero triple: Representa una matriz 2D dinámica donde cada celda almacena un puntero a un elemento (estación, jugador o espacio vacío). **Su uso es obligatorio**
- Struct definido en tablero.h

```
1 typedef struct {
2     void*** celdas;    // Matriz 2D de elementos (puntero triple
3                       // )
4     int filas;
5     int columnas;
6 } Tablero;
```

- Funciones en Tablero.c/h:

```
1 void inicializarTablero(Tablero* tablero, int filas, int
   columnas)
2 void mostrarTablero();
3 void actualizarCelda(Tablero* tablero, int x, int y, void*
   elemento)
```

Inventario

- Estructura: Puntero doble `void**`, que apunta a una lista estática de ingredientes y, eventualmente, el extintor.

- Funciones en `inventario.c/h`:

```
1 void crearInventario();
2 void agregarIngrediente();
3 void eliminarIngrediente();
4 void verInventario();
```

- Struct definido en `inventario.h`:

```
1 typedef struct {
2     char nombre[30];          // Ej: "pan", "pollo"
3     int estado;              // Ej: "0 = crudo", "1 = cortado", etc.
4     int es_extintor;          // 1 si es extintor, 0 si es ingrediente
5     int turnos_elaboracion;   // Turnos que tarda en prepararse.
6     int prob_incendio;        // Probabilidad de causar incendio.
7 } Ingrediente;
```

Estaciones y Acciones

- Estructura: Cada estación tendrá asociada una función a través de un puntero a función. Al interactuar con una estación, se invoca automáticamente la función correspondiente.

- Structs definidos en `acciones.h`

```
1 typedef struct {
2     char simbolo;              // 'T' (Tabla), 'C' (Cocina), 'A'
3     void (*accion)(void*, int, int); // Puntero a funcion (ej:
4     int en_llamas;             // 1 si hay incendio, 0 si no
5     int turnos_inhabilitada;    // Para estaciones apagadas con
6     } Estacion;
```

```
1 typedef struct {
2     char nombre_plato[50]      //Ej: McCharly
3     Ingrediente** ingredientes_requeridos; // Lista de
4     int completado;            // 1 si entregado, 0
5     } Pedido;
```

- Funciones clave (`acciones.h`) vinculadas a estaciones

```
1 void cortar(void* contexto, int x, int y); // Tabla de cortar
2 void cocinar(void* contexto, int x, int y); // Cocina
3 void buscar_ingrediente(void* contexto, int x, int y); //Almacen
4 void apagar_incendio(void* contexto, int x, int y); // Extintor
5 void entregar_pedido(void* contexto);
```