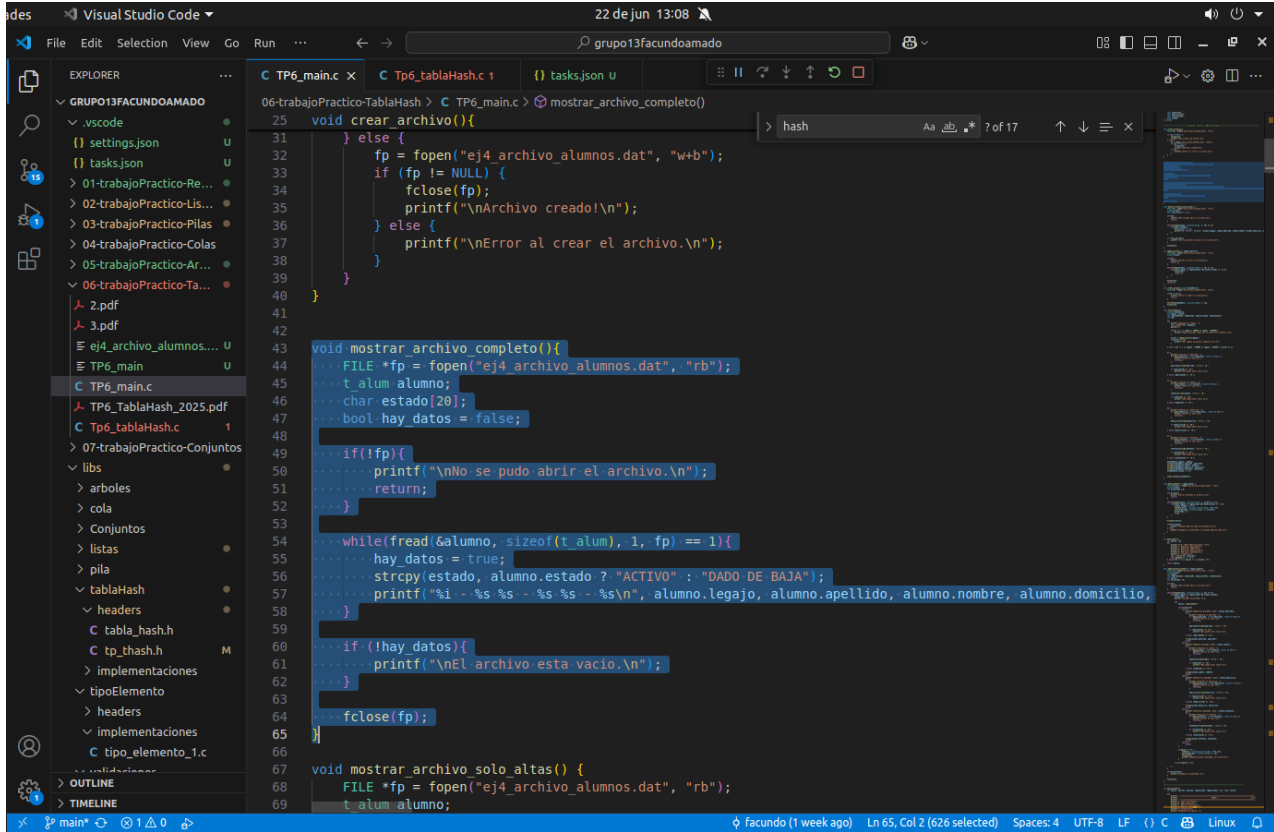


RESULTADO DE LA CORRECCIÓN: **APROBADO** —

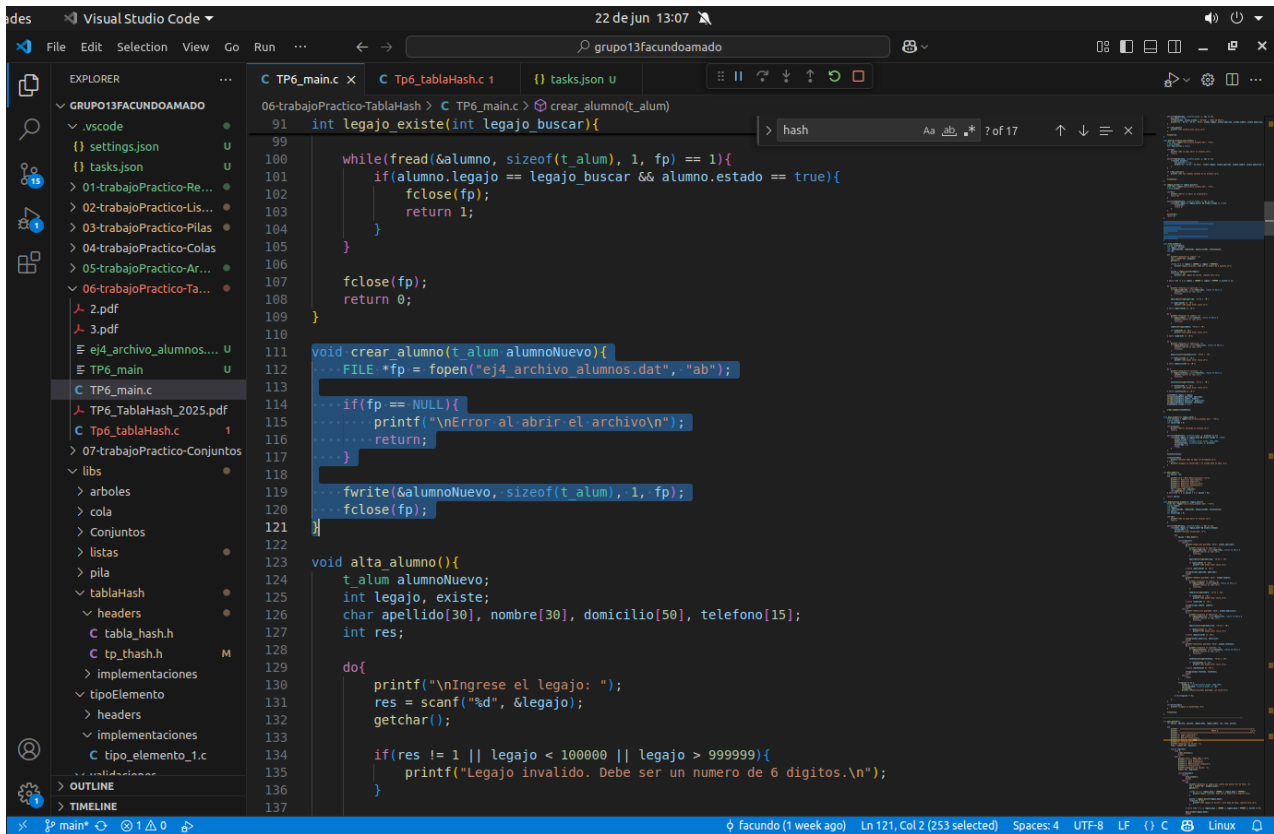
OBSERVACIONES

Ejercicio 4: resolución incorrecta. Solo llena la tabla hash si se ingresa al menú específico que la muestra. No lo hace cuando se da de alta el alumno. Tampoco la usa para listar. Muestra todo el contenido del archivo. No hay búsqueda por legajo.

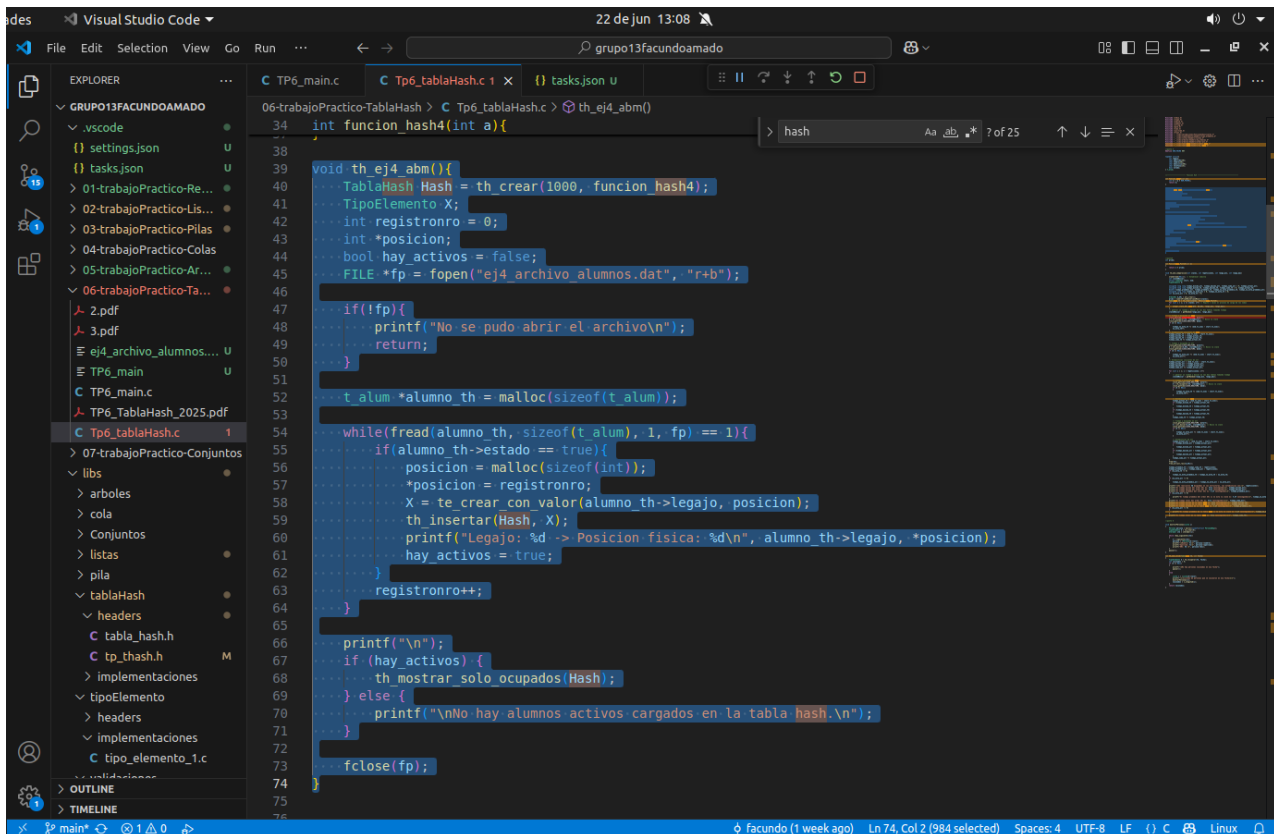


```
25 void crear_archivo(){
31 } else {
32     fp = fopen("ej4_archivo_alumnos.dat", "w+b");
33     if (fp != NULL) {
34         fclose(fp);
35         printf("\nArchivo creado!\n");
36     } else {
37         printf("\nError al crear el archivo.\n");
38     }
39 }
40
41
42 void mostrar_archivo_completo(){
43     FILE *fp = fopen("ej4_archivo_alumnos.dat", "rb");
44     t_alum alumno;
45     char estado[20];
46     bool hay_datos = false;
47
48     if(!fp){
49         printf("\nNo se pudo abrir el archivo.\n");
50         return;
51     }
52
53     while(fread(&alumno, sizeof(t_alum), 1, fp) == 1){
54         hay_datos = true;
55         strcpy(estado, alumno.estado ? "ACTIVO" : "DADO DE BAJA");
56         printf("%i - %s %s - %s %s - %s\n", alumno.legajo, alumno.apellido, alumno.nombre, alumno.domicilio,
57
58
59     if (!hay_datos){
60         printf("\nEl archivo esta vacio.\n");
61     }
62     fclose(fp);
63
64 void mostrar_archivo_solo_altas() {
65     FILE *fp = fopen("ej4_archivo_alumnos.dat", "rb");
66     t_alum alumno;
```

GRUPO 13 – Correcciones Trabajo Práctico: TABLAS HASH



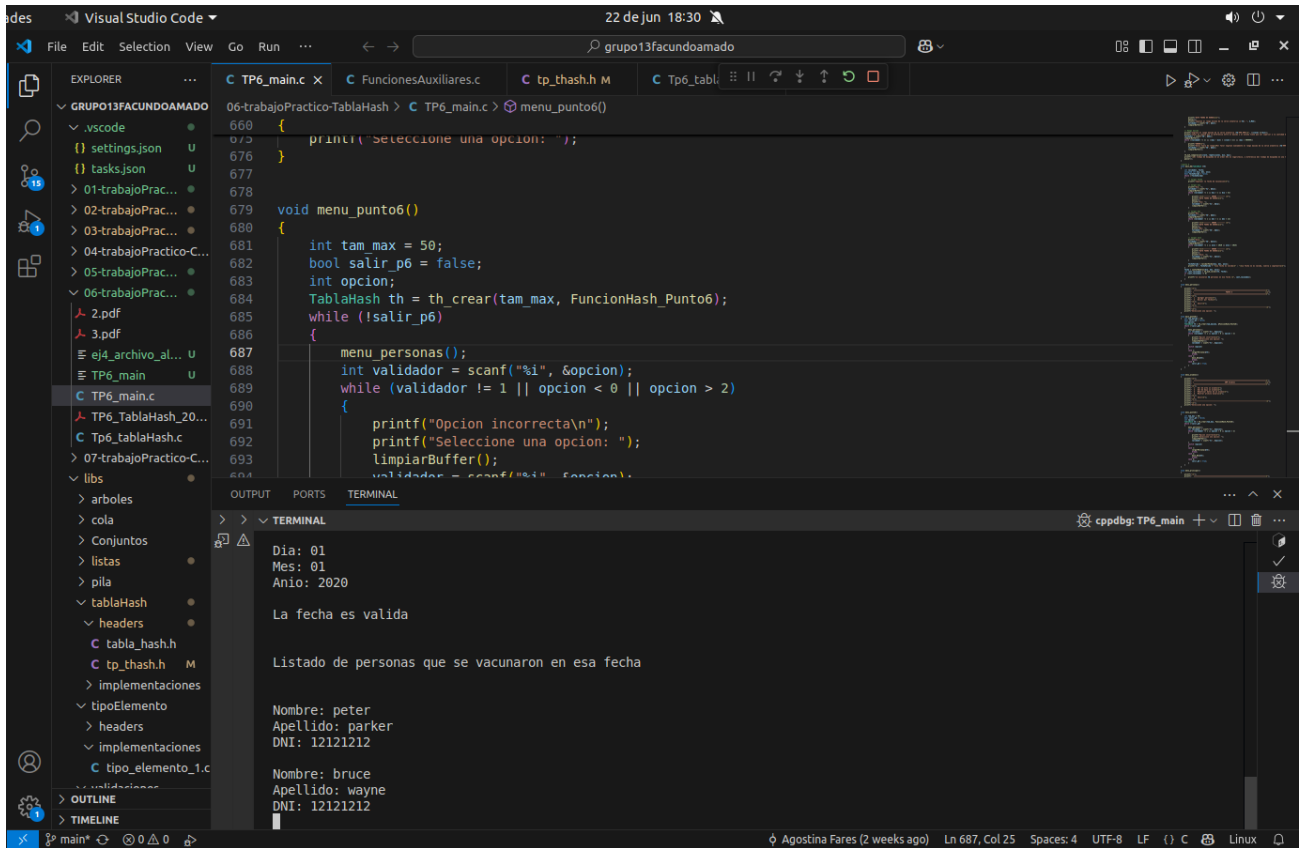
```
91 int legajo_existe(int legajo_buscar){
92     while(fread(&alumno, sizeof(t_alum), 1, fp) == 1){
93         if(alumno.legajo == legajo_buscar && alumno.estado == true){
94             fclose(fp);
95             return 1;
96         }
97     }
98     fclose(fp);
99     return 0;
100 }
101
102 void crear_alumno(t_alum alumnoNuevo){
103     FILE *fp = fopen("ej4_archivo_alumnos.dat", "ab");
104     if(fp == NULL){
105         printf("\nError al abrir el archivo\n");
106         return;
107     }
108     fwrite(&alumnoNuevo, sizeof(t_alum), 1, fp);
109     fclose(fp);
110 }
111
112 void alta_alumno(){
113     t_alum alumnoNuevo;
114     int legajo, existe;
115     char apellido[30], nombre[30], domicilio[50], telefono[15];
116     int res;
117     do{
118         printf("\nIngrese el legajo: ");
119         res = scanf("%d", &legajo);
120         getchar();
121         if(res != 1 || legajo < 100000 || legajo > 999999){
122             printf("Legajo invalido. Debe ser un numero de 6 digitos.\n");
123         }
124     } while(1);
125 }
```



```
34 int funcion_hash4(int a){
35     void th_ej4 abm(){
36         TablaHash Hash = th_crear(1000, funcion_hash4);
37         TipoElemento X;
38         int registraron = 0;
39         int *posicion;
40         bool hay activos = false;
41         FILE *fp = fopen("ej4_archivo_alumnos.dat", "r+b");
42         if(!fp){
43             printf("No se pudo abrir el archivo\n");
44             return;
45         }
46         t_alum *alumno_th = malloc(sizeof(t_alum));
47         while(fread(alumno_th, sizeof(t_alum), 1, fp) == 1){
48             if(alumno_th->estado == true){
49                 posicion = malloc(sizeof(int));
50                 *posicion = registraron;
51                 X = te_crear_con_valor(alumno_th->legajo, *posicion);
52                 th_insertar(Hash, X);
53                 printf("Legajo: %d -> Posicion fisica: %d\n", alumno_th->legajo, *posicion);
54                 hay activos = true;
55                 registraron++;
56             }
57         }
58         printf("\n");
59         if(hay activos){
60             th_mostrar_solo_ocupados(Hash);
61         } else {
62             printf("\nNo hay alumnos activos cargados en la tabla hash.\n");
63         }
64         fclose(fp);
65     }
66 }
```

GRUPO 13 – Correcciones Trabajo Práctico: TABLAS HASH

Ejercicio 6: permite cargar a la misma persona dos veces el mismo día de vacunación.



```
06-trabajoPractico-TablaHash > TP6_main.c > menu_punto6()
660 {
670     printf("Seleccione una opcion: ");
676 }
677
678 void menu_punto6()
679 {
680     int tam_max = 50;
681     bool salir_p6 = false;
682     int opcion;
683     TablaHash th = th_crear(tam_max, FuncionHash_Punto6);
684     while (!salir_p6)
685     {
686         menu_personas();
687         int validador = scanf("%i", &opcion);
688         while (validador != 1 || opcion < 0 || opcion > 2)
689         {
690             printf("Opcion incorrecta\n");
691             printf("Seleccione una opcion: ");
692             limpiarBuffer();
693             validador = scanf("%i", &opcion);
694         }
695     }
696 }
```

OUTPUT PORTS TERMINAL

cppdbg: TP6_main

Dia: 01
Mes: 01
Año: 2020

La fecha es valida

Listado de personas que se vacunaron en esa fecha

Nombre: peter
Apellido: parker
DNI: 12121212

Nombre: bruce
Apellido: wayne
DNI: 12121212