

a) 3: GPS, Trayecto, Localizacion.
 #define MAX_GPS 100
 typedef struct Localizacion {
 double latitud, longitud;
 } Localizacion;
 typedef struct Trayecto {
 Localizacion origen, destino;
 } Trayecto;
 typedef struct GPS {
 int n_trayectos;
 Trayecto * trayectos [MAX_GPS];
 } GPS;

b) Localizacion * crea Localizacion (double latitud, double longitud);
 Trayecto * crea Trayecto (Localizacion * origen, Localizacion * destino);
 Trayecto * copia Trayecto (Trayecto original);
 GPS * crea GPS();
 int get_Num_Trayectos_De_GPS (GPS * g);

c) status imprime_Trayectos_De_GPS (archivo f, GPS g) {
 g. trayectos = get_Num_Trayectos_De_GPS (g)
 imprime_Enter (f, g. trayectos)
 for (i=0; i < g. trayectos; ++i)
 imprime_Trayecto (f, g. trayectos [i])
 }

d) status anade_Trayecto_A_GPS (GPS * pGPS, const Trayecto * pTray) {
 int pGPS_trayectos = get_Num_Trayectos_De_GPS (pGPS);
 pGPS -> trayectos [pGPS_trayectos] = pTray;
 pGPS -> n_trayectos += 1;
 return OK;
 }

```

e) void main() {
    GPS *gps = creaGPS();
    Localizacion *ini, *fin;
    ini = crea Localizacion(34, 58);
    fin = crea Localizacion(41, 71);
    Trayecto *trayecto = crea Trayecto(ini, fin);
    AñadirTrayectoAlGPS(gps, trayecto);
    imprime Trayectos de GPS (std::cout, gps);
    elimina Trayectos de GPS (gps, trayecto);
    free(gps);
}
    
```