

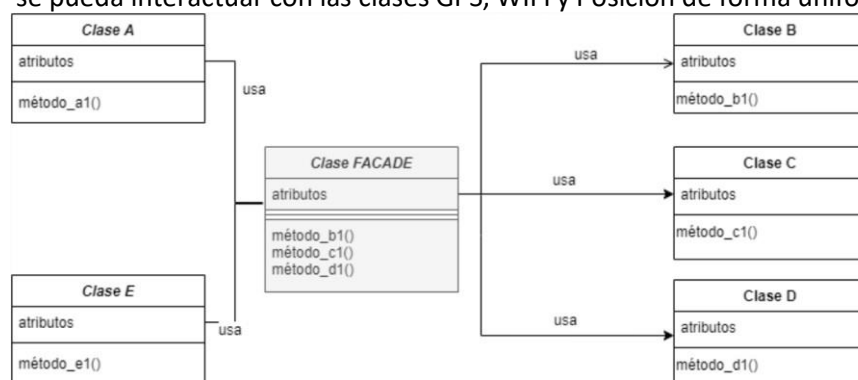
## ED\_T7\_Act6: Patrón Facade

Sobre el patrón **Facade** que hemos estudiado en Clase, responde a las siguientes cuestiones:

- ¿En qué situaciones está indicada la utilización de este Patrón de Diseño?

El **patrón facade** permite ocultar la complejidad de un sistema detrás de una interfaz más sencilla y fácil de usar

- Partiendo de la estructura de clases equivalente a la modelada dentro del siguiente diagrama, deberás implementar el patrón Facade de tal forma que desde el cliente se pueda interactuar con las clases GPS, WIFI y Posicion de forma uniforme:



```
/**
 * Chip del GPS
 * @author rhtuf
 */
public class GPS {

    public void Encender() {

        System.out.println("GPS encendido");

    }

    public void FijarPosicion() {

        System.out.println("Posición fijada");

    }

    public Posicion UltimaPosicion() {
        Posicion p = new Posicion();
        p.setLatitud("5.4");
        p.setLongitud("-0,5");
        p.setAltura("800");
        p.setFecha("2021--04-02 20:00");
        return p;
    }

    public void Apagar() {
        System.out.println("Apagar GPS");
    }

}
```

```
* @author rhtuf
*/
public class Posicion {

    private String Latitud;
    private String Longitud;
    private String Altura;
    private String Fecha;

    public String getLatitud() {
        return Latitud;
    }

    public void setLatitud(String Latitud) {
        this.Latitud = Latitud;
    }

    public String getLongitud() {
        return Longitud;
    }

    public void setLongitud(String Longitud) {
        this.Longitud = Longitud;
    }

    public String getAltura() {
        return Altura;
    }

    public void setAltura(String Altura) {
        this.Altura = Altura;
    }

    public String getFecha() {
        return Fecha;
    }

    public void setFecha(String Fecha) {
        this.Fecha = Fecha;
    }

    @Override
    public String toString() {
        return "Posicion{" + "Latitud=" + Latitud + ", Longitud=" + Longitud + ", Altura=" + Altura + ", Fecha=" + Fecha + '}';
    }
}
```

```
/**
 * Adaptador WIFI
 * @author rhtuf
 */
public class Wifi {

    private String wifi;

    public void Encender() {
        System.out.println("WIFI encendida");
    }

    public void Conectar(String SSID) {

        wifi = SSID;
        System.out.println("Conectar a WIFI " + SSID);

    }

    public void Desconectar() {
        System.out.println("Desconectar WIFI " + wifi);
    }

    public void Apagar() {
        System.out.println("Wifi apagada");
    }

    public String getWifi() {
        return wifi;
    }

    public void setWifi(String wifi) {
        this.wifi = wifi;
    }
}
```

- **Tareas Entregables:**
- Creación de una nueva clase 'Facade.java' que implemente la funcionalidad de Fachada.
- Habrá que modificar la clase Cliente.java para que utilicen la nueva clase 'Facade.java'. Esta clase contendrá la siguiente funcionalidad:
  - ConectarAWifi al que se le pasará el nombre de la wifi a conectar
  - DesconectarDeWifi
  - ObtenerPosicion devolverá la última posición por GPS después de ser fijada
- Incluye en un documento PDF la captura de la clase Fachada y una captura de la salida de la ejecución dicha clase desde la clase Cliente, que será similar a: