

Asignatura.java

```
public Asignatura() {  
    this(nombre:"Sin nombre",Tipo.FUNDAMENTACION);  
    this(0);  
}
```

El this que está en el constructor debe estar en la primera línea, por lo que es mejor eliminarlo ya que no interfiere en el resultado del código

```
public Asignatura(int codigoInterno,Tipo tipo) {  
    this(nombre:"Sin nombre", codigoInterno, codigoExterno:0,tipo);  
}  
  
public Asignatura(int codigoExterno,Tipo tipo) {  
    this(nombre:"Sin nombre", codigoInterno:0, codigoExterno,tipo);  
}
```

Ambos constructores están recibiendo el mismo tipo de dato y esto genera ambigüedad, para solucionar este problema se pueden unificar estos dos constructores en uno solo para que reciban codigoInterno y codigoExterno en un mismo constructor o la otra opción que es la que va a implementar es dejar solo uno de los constructores, para este caso, dejaremos el constructor Asignatura(int codigoExterno,Tipo tipo)

```
public void cambiarDatos(int codigoInterno) {  
    this.codigoInterno = codigoInterno;  
}  
  
public void cambiarDatos(int codigoExterno) {  
    this.codigoExterno = codigoExterno;  
}
```

En este caso, pasa lo mismo que el error anterior, debido a esto se debe decidir cuál método se deja, debido a que este método funciona como un set y ya hay un set para código interno, entonces se deja el método cambiarDatos(int codigoExterno)

```
public Grupo(Persona[] estudiantes, Persona profesor, Asignatura asignatura, int codigo, String horario) {
    this.estudiantes = estudiantes;
    this.profesor = profesor;
    this.asignatura = asignatura;
    this.codigo = codigo;
    this.horario = horario;
}
```

this.codigo es una variable inmutable por lo que no se puede modificar su valor por lo que se elimina el decorador final

```
public Grupo(int cantidadEstudiantes, Persona profesor, Asignatura asignatura, int codigo, String horario) {
    Persona[] personas = new Persona[cantidadEstudiantes];
    this(personas, profesor, asignatura, codigo, horario);
}
```

El constructor debe ser llamado en la primer linea, entonces se elimina la variable personas y se crea directamente en el constructor

```
static {
    totalPersonas = 0;
    cedula = 3;
}
```

La variable cédula debe ser estatica para que no presente el error pero en el contexto en el que se presenta, es mejor eliminarla

```
public Persona(String nombre) {
    this.nombre = "";
    totalPersonas++;
}
```

Se debe agregar una linea que inicialice el valor de cédula

```
public enum Tipo {
    DISCIPLINAR(10), FUNDAMENTACION(20), ELECTIVA(30);

    private int codigo;
    private String nombre;
    public Tipo(int codigo, String nombre) {
        this.codigo = codigo;
        this.nombre = nombre;
    }
}
```

El error se debe a que el constructor pide código y nombre y cuando se crea el tipo, solo se está enviando el código