

Cambios realizados en el esquema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.uniovi.es"
xmlns="http://www.uniovi.es"
elementFormDefault="qualified" >
```

Cambio en el encoding para que sea el UTF-8.

Creación de un name space

Modificación del xmlns al indicado en el cambio.

```
<xs:element name="longitudCarrera">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:float">
        <xs:attribute name="unidad" type="xs:string" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="anchuraMedia">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:float">
        <xs:attribute name="unidad" type="xs:string" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

Cambio en el tipo de longitud carrera y anchura media de String a float porque este dato es un número con decimales y es más correcto que el tipo String, además con el float se podrían realizar validaciones numéricas(se pasa de un ancho o una longitud o se queda corto)

```

<xs:element name="horarioCircuito">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="fechaCarrera" />
      <xs:element ref="horaCarrera" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="fechaCarrera" type="xs:date" />
<xs:element name="horaCarrera" type="xs:time" />
<xs:element name="datosCarrera">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="numeroVueltas" />
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

Cambio en el tipo de fecha carrera a date(aaaa-mm-dd) porque es una fecha y facilita comprobaciones entre fechas.

Cambio en el tipo de hora carrera a time (horas: minutos: segundos) porque es un hora y facilita las comprobaciones entre horas y tiempos

```

<xs:element name="numeroVueltas" type="xs:integer" />

```

Cambio en el tipo de numero vueltas de String a integer, porque es un número entero (nunca va a tener decimales) y facilita comprobaciones (se establece un máximo o mínimo de vueltas).

```

<xs:element name="referencias">
  <xs:complexType>
    <xs:element name="videos">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="3" ref="video" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="video">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="enlace" />
        </xs:sequence>
        <xs:attribute name="descripcion" type="xs:string" use="required" />
      </xs:complexType>
    </xs:element>
  </xs:complexType>
</xs:element>

```

Cambio en referencias para que se adecue a lo requerido, tiene que haber un mínimo de 3 referencias y sin límite máximo

```

<xs:element name="imagenes">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="5" ref="imagen" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="imagen">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="enlace" />
    </xs:sequence>
    <xs:attribute name="descripcion" type="xs:string" use="required" />
  </xs:complexType>
</xs:element>

```

Cambio en imagen para que se adecue a lo requerido, tiene que haber un mínimo de 1 imagen y un máximo de 5

```

<xs:element name="videos">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="3" ref="video" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="video">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="enlace" />
    </xs:sequence>
    <xs:attribute name="descripcion" type="xs:string" use="required" />
  </xs:complexType>
</xs:element>

```

Cambio en video para que se adecue a lo requerido, tiene que haber un mínimo de 0 videos y un máximo de 3

```

<xs:simpleType name="LongitudType">
  <xs:restriction base="xs:float">
    <xs:minInclusive value="-180.0"/>
    <xs:maxInclusive value="180.0"/>
  </xs:restriction>
</xs:simpleType>

<xs:element name="longitud">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="LongitudType">
        <xs:attribute name="unidad" type="xs:string" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

```

Cambio en longitud, se cambia su tipo a LongitudType que es de tipo float (las coordenadas están en formato dd (decimal degree) que es un número con decimales ), se crea este LongitudType porque es un tipo simple que es en el que se puede realizar comprobaciones como es en este caso que la longitud este entre -180 y 180 que serían los posibles valores que tiene.

```

<xs:simpleType name="LatitudType">
  <xs:restriction base = "xs:float">
    <xs:minInclusive value= "-90"/>
    <xs:maxInclusive value= "90"/>
  </xs:restriction>
</xs:simpleType>

<xs:element name="latitud">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="LatitudType">
        <xs:attribute name="unidad" type="xs:string" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

```

Cambio en latitud, se cambia su tipo a LatitudType que es de tipo float (las coordenadas están en formato dd (decimal degree) que es un número con decimales), se crea este LatitudType porque es un tipo simple que es en el que se puede realizar comprobaciones como es en este caso que la latitud este entre -90 y 90 que serían los posibles valores que tiene.

```

<xs:element name="altitud">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:float">
        <xs:attribute name="unidad" type="xs:string" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

```

Cambio en el tipo de altitud de String a float como es un número que puede o no tener decimales se deja en tipo float.

```

<xs:element name="distancia">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:float">
        <xs:attribute name="unidad" type="xs:string" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="numeroSector" type="xs:integer" />

```

Cambio en el tipo de distancia (entre dos tramos) a float, porque es un número con decimales y en el tipo de numero sector de String a integer, porque el sector se indica con un número natural.