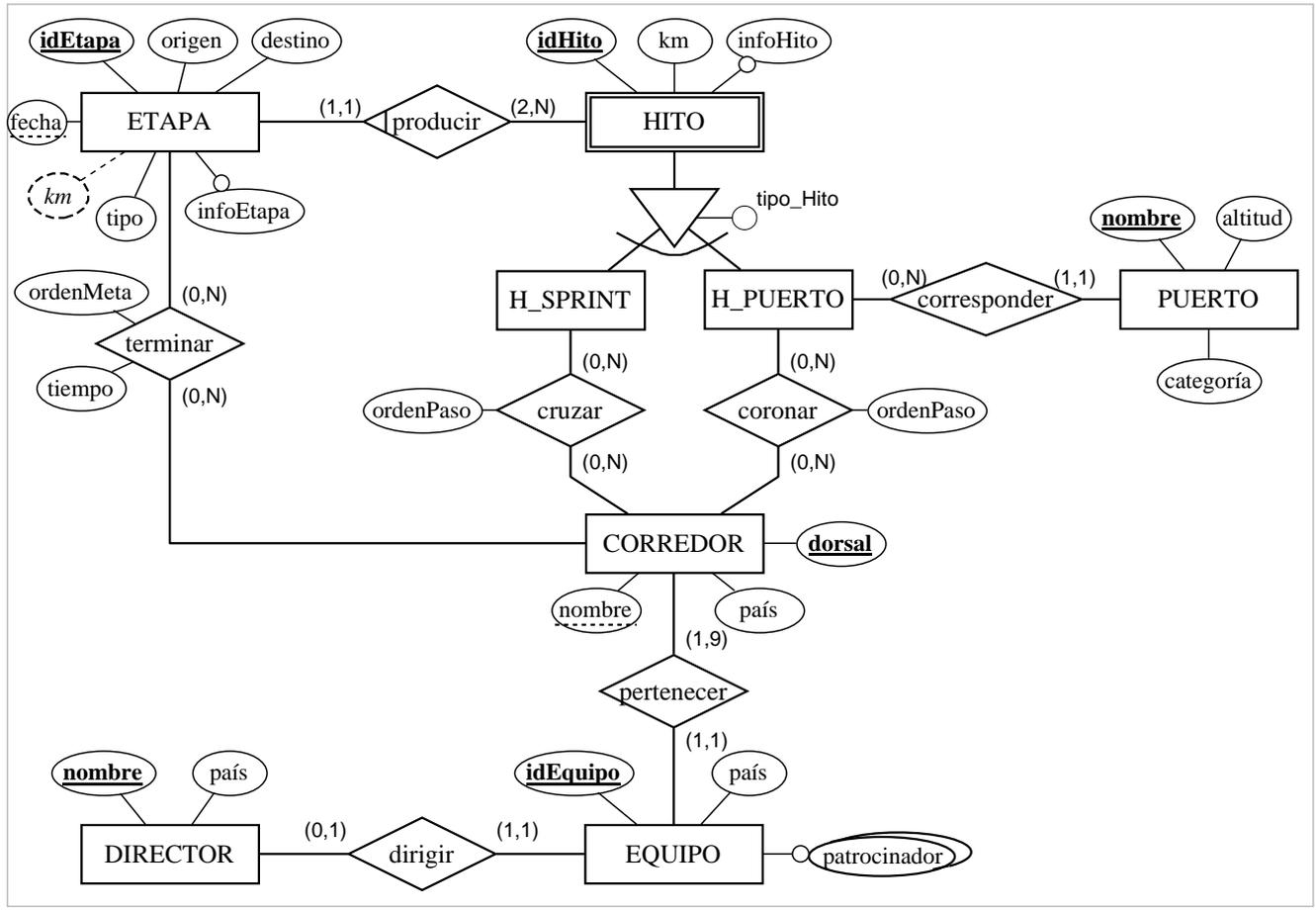


**Esquema E/R de la de la base de datos de la Vuelta Ciclista**



**DOMINIOS:**

tpNombre = cadena(30);  
 tpSecuencia = natural, autoincremento;  
 tpEtapa = (llana, montaña, contrarreloj\_Equipos,...);  
 tpHito = (Informacion, Comienzo, Meta, Puerto, Sprint, Avituallamiento);  
 tpPais = (España, Francia, Alemania, Italia, Holanda, Dinamarca, Portugal, Inglaterra);

**ATRIBUTOS:**

dorsal, idEtapa, idHito, ordenMeta, ordenPaso : tpSecuencia;  
 origen, destino, nombre, idEquipo, patrocinador : tpNombre;  
 pais : tpPais;  
 km, altitud : real;  
 tiempo : tpHora; {hora, minuto, segundo, décima, o mejor un número entero (o real) con el nº total de segundos}

**RESTRICCIONES:**                      •••• (sólo alguna a modo de ejemplo)

- \* verificar que ∇ ocurrencia de terminar de una etapa, ordenMeta ES ÚNICO.
- \* verificar que ∇ ocurrencia de cruzar de un hito de sprint, ordenPaso ES ÚNICO.
- \* verificar que ∇ ocurrencia de coronar de un hito de puerto, ordenPaso ES ÚNICO.

**Esquema relacional de la base de datos de la Vuelta Ciclista**

**DOMINIOS** (los expresados en el esquema E/R)

**ESQUEMAS DE RELACION**

**ETAPA** (  
idEtapa : tpNatural; -- autoincrementado(1..N)  
 origen, destino : tpNombre, **NO NULO**; fecha : tpFecha, **NO NULO, UNIQUE**;  
 tipo : tpEtapa, **NO NULO**; km : tpReal; info : tpInfo);

**HITO** (  
idEtapa : tpNatural, **clave ajena de ETAPA**, borrado en CASCADA;  
idHito : tpNatural; -- autoincrementado(1..N) para cada idEtapa  
 tipoHito : tpHito, **NO NULO**;  
 km : tpReal, **NO NULO**; {distancia desde el origen de la etapa}  
 info : tpInfo;  
 idPuerto : tpNombre, **clave ajena de PUERTO**, borrado en CASCADA);  
 \* verificar que idPuerto es **NO NULO** sii tipoHito=puerto

**CORREDOR** (  
dorsal : tpNatural;  
 nombre : tpNombre, **NO NULO, UNICO**;  
 pais : tpNombre, **NO NULO**;  
 idEquipo : tpNombre, **NO NULO, clave ajena de EQUIPO**, borrado en CASCADA);

**LLEGADA** (  
idEtapa: tpNatural, **clave ajena de ETAPA**, borrado en CASCADA;  
dorsal : tpNatural, **clave ajena de CORREDOR**, borrado en CASCADA;  
 ordenMeta : tpNatural, **NO NULO**; -- autoincrementado(1..N) para cada idEtapa  
 tiempo : tpDuracion, **NO NULO**;  
 (idEtapa, ordenMeta), **UNICO**);  
 \* verificar que  $\forall$  ordenMeta<sub>i</sub> < ordenMeta<sub>j</sub>  $\rightarrow$  tiempo<sub>i</sub> <= tiempo<sub>j</sub>

**SPRINT** (  
idEtapa: tpNatural;  
idHito : tpNatural;  
dorsal : tpNatural, **clave ajena de CORREDOR**, borrado en CASCADA;  
 ordenPaso : tpNatural, **NO NULO** -- autoincrementado(1..N) para cada (idEtapa,idHito);  
 (idEtapa, idHito), **clave ajena de HITO**, borrado en CASCADA;  
 (idEtapa, idHito, ordenPaso), **UNICO**);  
 \* verificar que HITO[idEtapa,idHito].tipoHito=sprint

**CORONAR** (  
idEtapa: tpNatural;  
idHito : tpNatural;  
dorsal : tpNatural, **clave ajena de CORREDOR**, borrado en CASCADA;  
 ordenPaso : tpNatural, **NO NULO** -- autoincrementado(1..N) para cada (idEtapa,idHito);  
 (idEtapa, idHito), **clave ajena de HITO**, borrado en CASCADA;  
 (idEtapa, idHito, ordenPaso), **UNICO**);  
 \* verificar que HITO[idEtapa,idHito].tipoHito=puerto

**PUERTO** (  
idPuerto : tpNombre;  
 categoría : tpCategoriaP, **NO NULO**;  
 altitud : tpReal, **NO NULO**);

**EQUIPO** (  
nombEquipo : tpNombre;  
 pais : tpNombre, **NO NULO**;  
 director : tpNombre, **clave ajena de DIRECTOR**, borrado PUESTA A NULO);

**PATROCINIO** (  
nombEquipo : tpNombre, **clave ajena de EQUIPO**, borrado en CASCADA;  
patrocinador : tpNombre);

**DIRECTOR** (  
nombre : tpNombre;  
 pais : tpNombre, **NO NULO**);

**RESTRICCIONES** •••• (sólo alguna, a modo de ejemplo)

\* verificar que el número de corredores de un equipo es <= 9 en todo momento.