

Universidad ORT Uruguay
Facultad de Ingeniería
Ing. Bernard Wand-Polak

Base de datos 2

Obligatorio 1

Pablo Pereira - 210162
Marcelo Rezzano - 195348

Entregado como requisito de la materia Base de datos 2

29 de junio de 2020

Índice

1 - Aclaraciones	2
2 - Introducción	2
3 - Supuestos	4
4 - Restricciones estructurales	5
5 - Restricciones no estructurales	7
6 - Procedures	8

1 - Aclaraciones

- Pablo Pereira (210162) es alumno del grupo de Enrique Latorres.
- Marcelo Rezzano (195348) es alumno del grupo de Valeria Emanuelli.

2 - Introducción

Para esta primera entrega decidimos comenzar por la creación de un MER según los datos obtenidos de la letra.

A partir de esto logramos idealizar y visualizar las distintas tablas que conformarán nuestra solución y serán utilizadas para la siguiente instancia.

Realizamos diversas modificaciones buscando siempre mejorar nuestro modelo y al fin obtuvimos como resultado el siguiente MER:

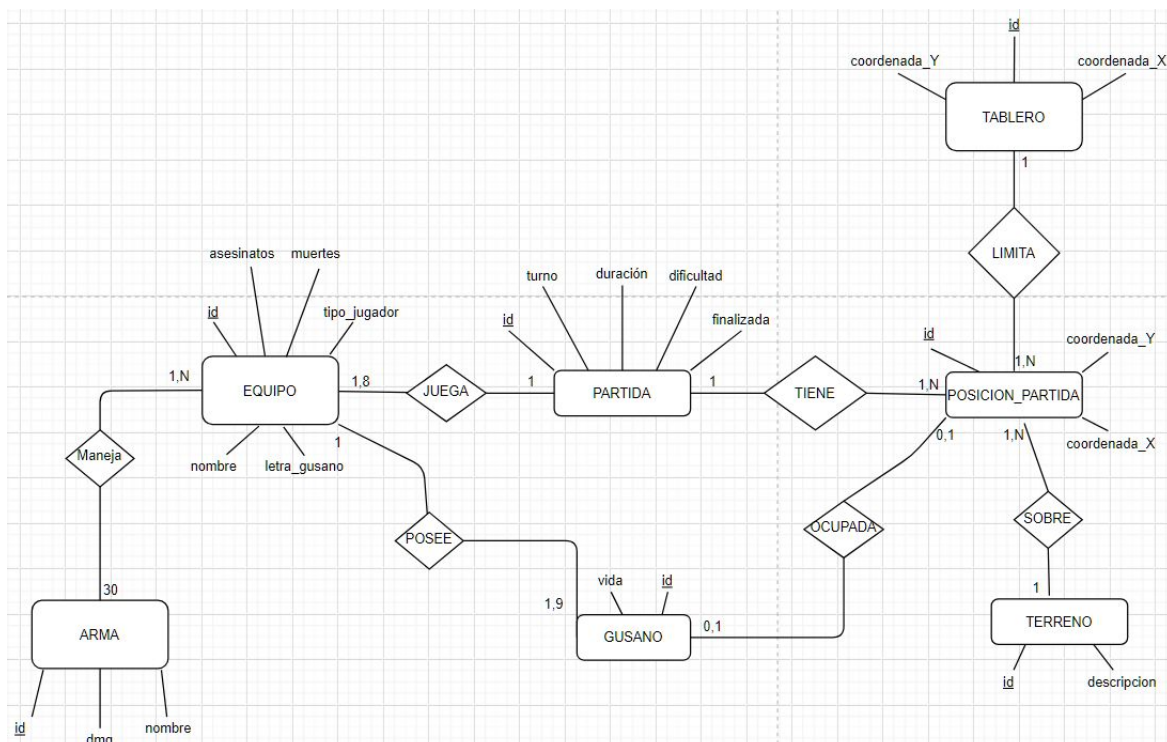


Figura 1 - Modelo Entidad Relación de la solución.

Siguiendo el MER se obtuvieron las siguientes tablas:

- La tabla ARMA consiste de un **identificador**, el **daño** que genera y su **nombre**.
- La tabla EQUIPO consiste de un **identificador**, su **nombre**, la cantidad de **asesinatos**, la cantidad de **muerres** del equipo y **una letra** que representa al equipo. Esta tabla también posee el atributo **tipo de jugador** que indica si es controlada por inteligencia artificial o un ser humano y un **identificador de partida** indicando a cuál partida pertenece.
- La tabla PARTIDA consiste de un **identificador**, la **dificultad** de la misma y su **duración** final. A su vez la tabla posee un indicador **finalizada** para saber si la misma ha terminado y el atributo **turno** que indica cual es el equipo a realizar la jugada.
- La tabla GUSANO consiste de un **identificador**, la cantidad de **vida** que posee y un **identificador de equipo** indicando a cuál equipo pertenece.
- La tabla TERRENO consiste de un **identificador** y una **descripción**.
- La tabla TABLERO consiste de un **identificador**, una **coordenada X** y una **coordenada Y** que representan las celdas del tablero.
- La tabla POSICION_PARTIDA consiste de un **identificador**, una **coordenada X** y una **coordenada Y**. A su vez también posee un **identificador de partida** indicando a cuál partida pertenece, un **identificador de gusano** y un **identificador de equipo** para saber cuál es el gusano que está en dicha posición si es que existe, un **identificador de terreno** para conocer cuál terreno está situado en la misma y por último un **identificador de tablero**.

3 - Supuestos

1. Un mismo equipo solo puede participar de una partida.
2. Un mismo gusano solo puede participar de un equipo.
3. Un mismo equipo no puede repetir un arma.
4. Para representar los campos booleanos decidimos utilizar un número, siendo **0** el representante de **false** y **1** el representante de **true**.
5. Decidimos no utilizar el auto increment ya que en algunas versiones no existe.
6. Todas las armas actúan iguales, producen el mismo daño al enemigo menos el burro.

4 - Restricciones estructurales

A continuación se listaran todas las restricciones estructurales que fueron encontradas.

TABLA	NOMBRE	INFORMACIÓN
TABLERO	ID, COORDENADA_X, COORDENADA_Y	PRIMARY KEY
TABLERO	COORDENADA_X	$0 < x < 51$
TABLERO	COORDENADA_Y	$0 < y < 16$
PARTIDA	ID	PRIMARY KEY
PARTIDA	DIFICULTAD	('DIFICIL', 'INTERMEDIO', 'FACIL')
PARTIDA	DURACION	('15', '20', '25')
TERRENO	ID	PRIMARY KEY
TERRENO	TIPO	('T', 'A', '.', 'P', 'B')
EQUIPO	ID	PRIMARY KEY
EQUIPO	TIPO_JUGADOR	('H', 'M')
EQUIPO	PARTIDAID	FOREIGN KEY
GUSANO	ID	PRIMARY KEY
GUSANO	EQUIPOID	FOREIGN KEY
GUSANO	VIDA	$0 < > 100$
POSICION_PARTIDA	ID, COORDENADA_X, COORDENADA_Y, PARTIDAID	PRIMARY KEY
POSICION_PARTIDA	PARTIDAID	FOREIGN KEY
POSICION_PARTIDA	EQUIPOID, GUSANOID	FOREIGN KEY
POSICION_PARTIDA	GUSANOID	FOREIGN KEY
POSICION_PARTIDA	TABLEROID, COORDENADA_X,	FOREIGN KEY

	COORDENADA_Y	
ARMA	ID	PRIMARY KEY
ARMA	DMG	'100'
ARMA_EQUIPO	ID, ARMAID, EQUIPOID	PRIMARY KEY
ARMA_EQUIPO	ARMAID	FOREIGN KEY
ARMA_EQUIPO	EQUIPOID	FOREIGN KEY

5 - Restricciones no estructurales

A continuación se listaran todas las restricciones no estructurales que fueron encontradas.

RESTRICCIÓN	TIPO
UN EQUIPO TIENE HASTA 8 GUSANOS	TRIGGER
UN EQUIPO TIENE HASTA 30 ARMAS	TRIGGER
UNA PARTIDA PUEDE TENER HASTA 4 EQUIPOS (*)	TRIGGER
AL MENOS UN JUGADOR HUMANO POR PARTIDA (*)	TRIGGER
SOLO SE PUEDE INSERTAR UN GUSANO EN UNA POSICIÓN DE TIPO AIRE CON UNA POSICIÓN DE TIPO TIERRA POR ABAJO (*)	TRIGGER
A UN EQUIPO SOLO SE LE PUEDE ASIGNAR UNA PARTIDA SI POSEE ENTRE 1 y 30 ARMAS (*)	TRIGGER

Durante el desarrollo de los triggers nos hemos enfrentado al error:

'oracle ora-04091 table is mutating trigger/function may not see it'

Sin embargo, debido a restricciones de tiempo no pudimos finalizar la implementación de la solución del mismo, lo cual se basa en realizar triggers INSTEAD OF INSERT que se apoyan en vistas para su funcionamiento.

Se han creado las vistas pero se mantuvo la implementación que corre en los demás DBMS pero no para *Oracle*.

Los triggers marcados con (*) en la tabla mencionada anteriormente son los que muestran dicha excepción.

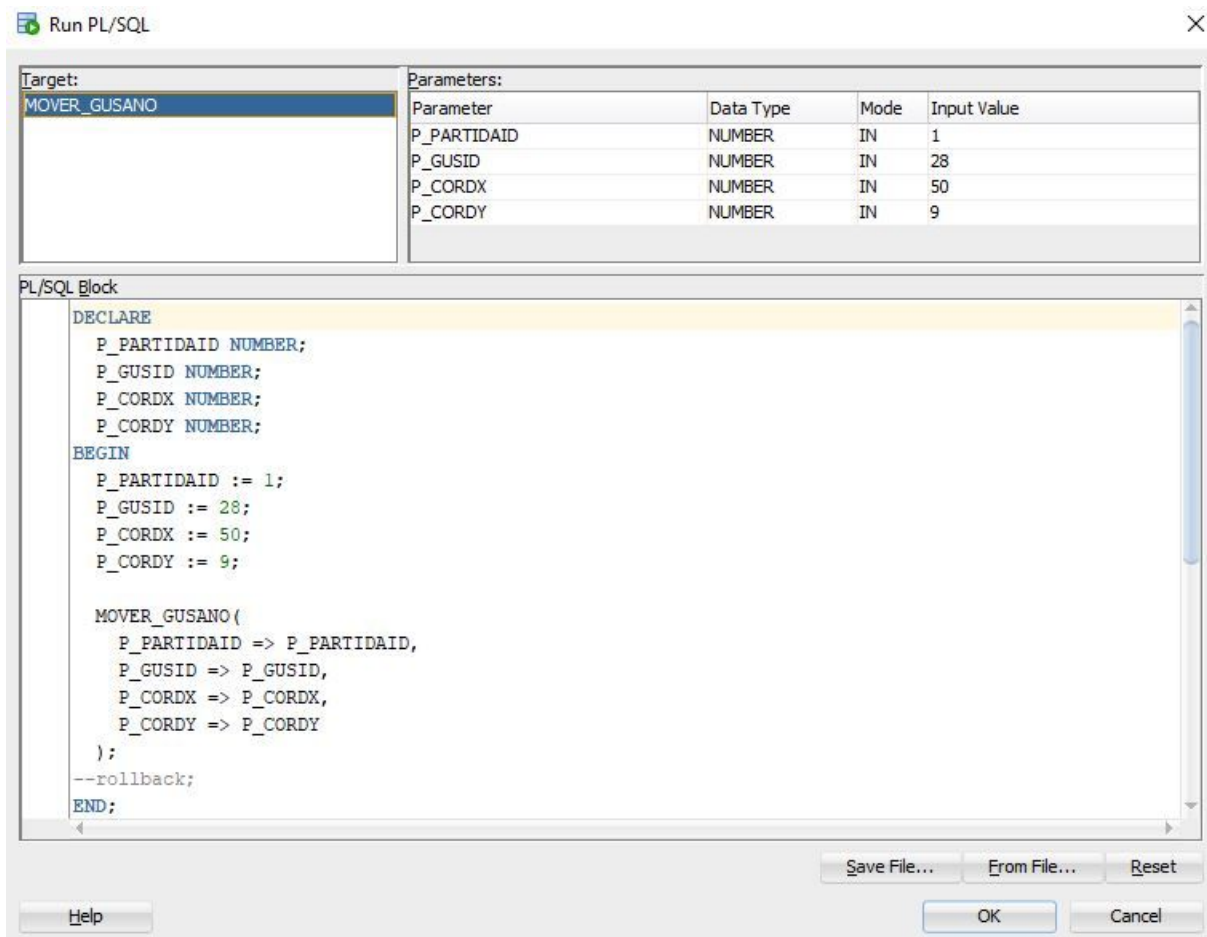


Figura 5.2.2 - Ejecutar SP Oscilar por la cuerda.

3. ¡Suelta un burro!: Proveer un servicio que dada una posición horizontal, en el contexto de una partida, “suelta” el arma del burro.

```
Connecting to the database OBL2.
TABLERO ANTES DEL BURRO:
//////////////////////////////////
.....
.....
.....
.....
.....W.....W.....R.....H.....
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
.....
.....W.....H.....
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
//////////////////////////////////
TABLERO DESPUES DEL BURRO:
.....
.....
.....
.....
.....W.....R.....H.....
TT....TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TT....TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
.....
.....H.....
TT....TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
TT....TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Process exited.
Disconnecting from the database OBL2.
```

Figura 5.3.1 - Salida SP Suelte un burro.

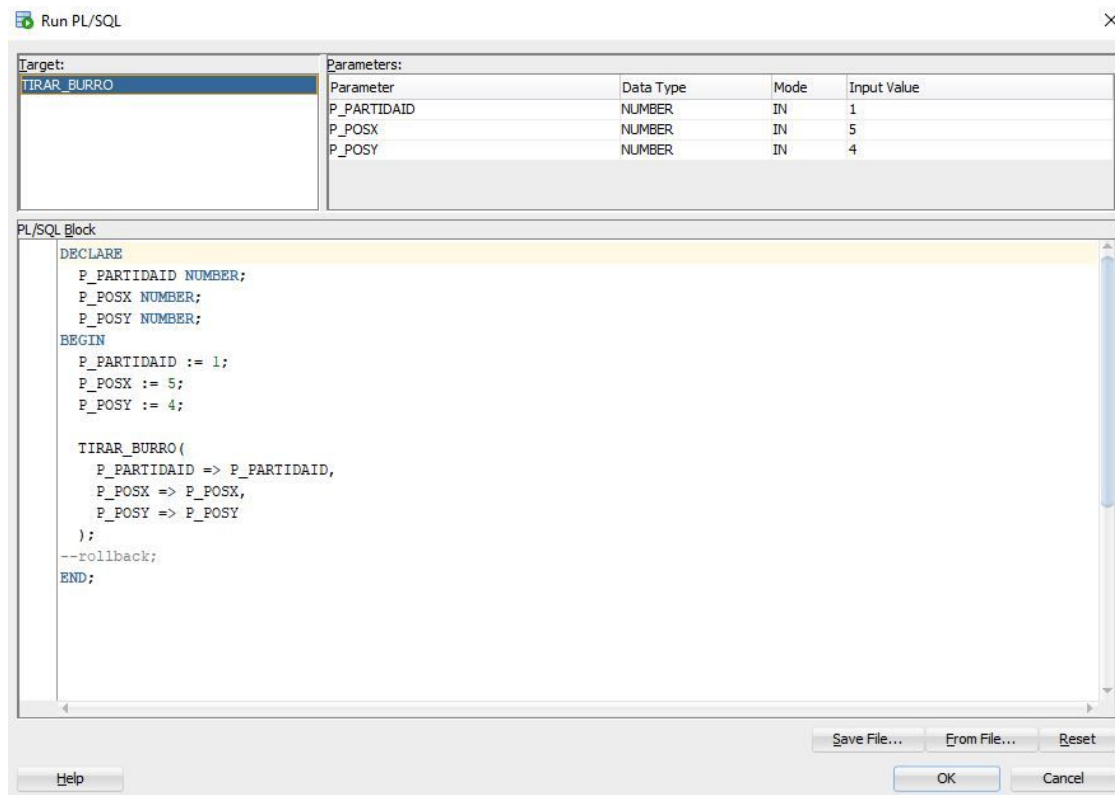


Figura 5.3.2 - Ejecutar Suelto un burro.

4. Resumen de la partida: Proveer un servicio que retorne el resumen de la partida, que será invocado por la aplicación cuando finalice.

```
Connecting to the database OBL2.  
NOMBRE JUGADOR = EQUIPO D, ASESINATOS = 7, MUERTES = 2  
NOMBRE JUGADOR = EQUIPO A, ASESINATOS = 5, MUERTES = 3  
NOMBRE JUGADOR = EQUIPO C, ASESINATOS = 3, MUERTES = 5  
NOMBRE JUGADOR = EQUIPO B, ASESINATOS = 2, MUERTES = 7  
Process exited.  
Disconnecting from the database OBL2.
```

Figura 5.4.1 - Salida SP Resumen de la partida.

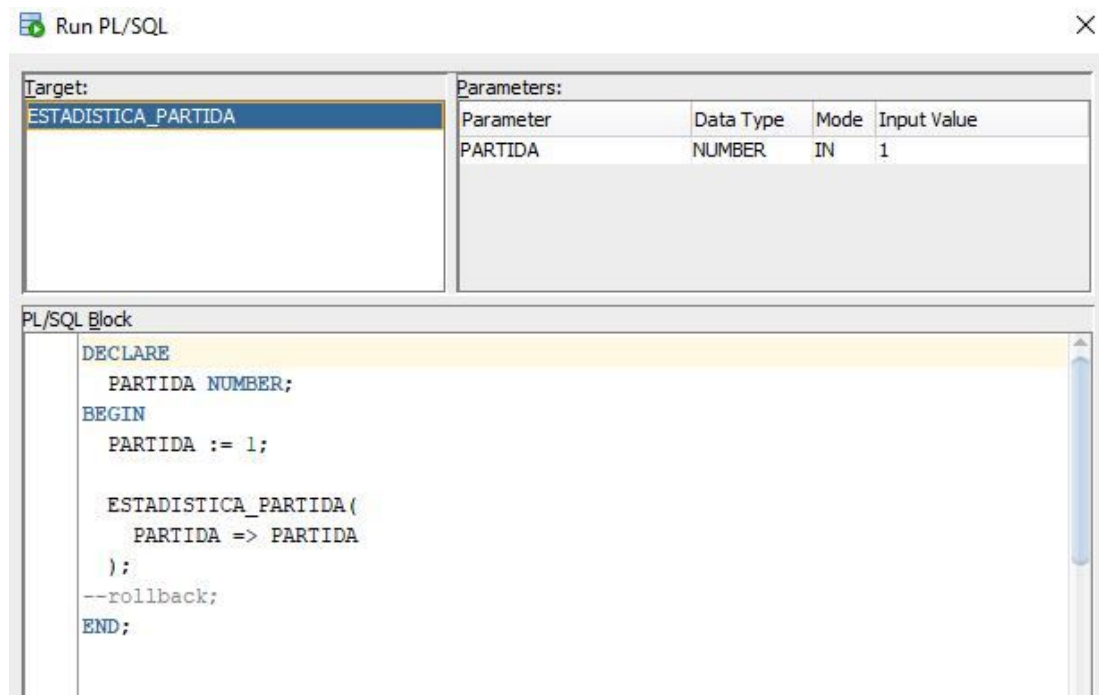


Figura 5.4.2 - Ejecutar SP Resumen de la partida.

5. Eliminar una partida: Proveer un servicio que reciba por parámetro una partida y la elimine de modo que no quede registro de sus datos relacionados, como si no hubiera existido.

[illegible]

Figura 5.5.1 - Salida SP Eliminar una partida.

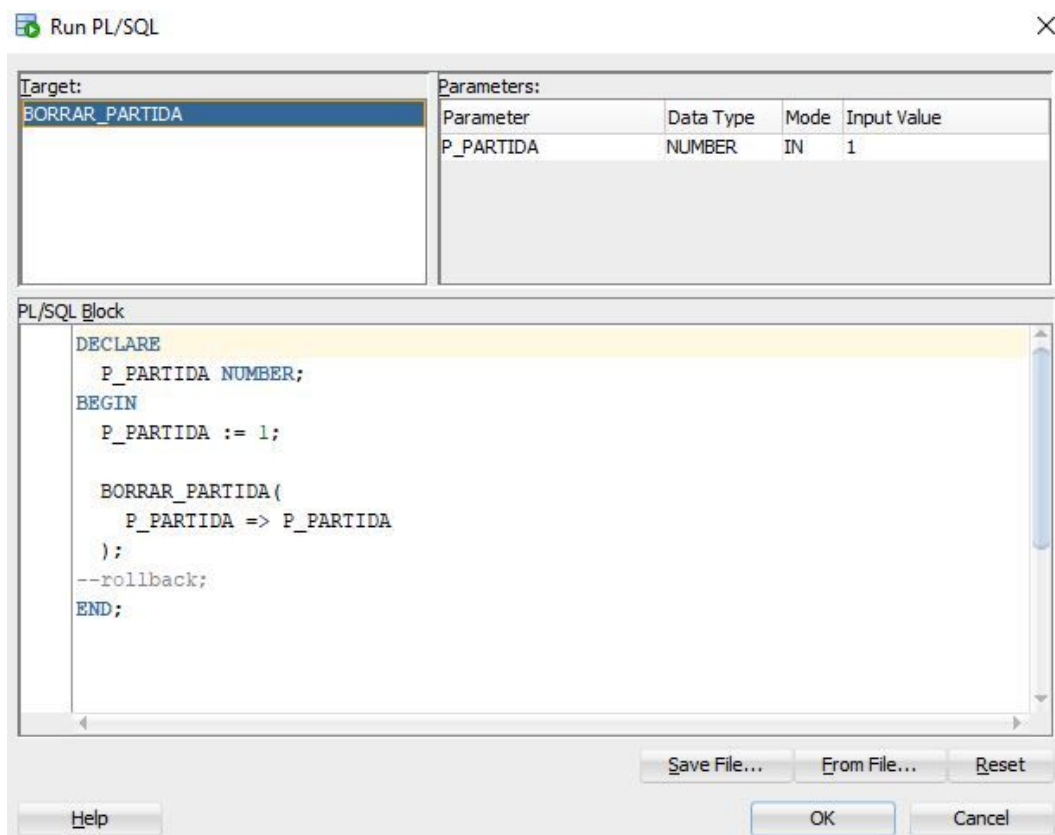


Figura 5.5.2 - Ejecutar SP Eliminar una partida.