

 <b>Proyecto Cupí2</b>	<b>ISIS-1204 Algorítmica y Programación</b> <b>Descripción</b>
Ejercicio:	N12_Continental
Autor:	Equipo Cupí2 2014-10
Fecha:	Abril 22 de 2014

## Enunciado

El objetivo del ejercicio es construir un programa distribuido que permita a un grupo de personas jugar en parejas el juego “Continental”, desde diferentes computadores. Como un servicio adicional, el programa debe almacenar información sobre los juegos perdidos y ganados por cada uno de los participantes.

El juego se desarrolla usando una baraja inglesa: 52 cartas distribuidas en 4 palos (picas, corazones, diamantes y tréboles) siendo numeradas del 1 (o as) al 10, seguidas por J, Q y K. Al iniciar el juego se barajan las cartas de manera aleatoria y se decide cuál de los dos jugadores empezará la partida. Posteriormente, se reparte un total de siete cartas a cada jugador y se muestra la primera carta de la baraja.

El objetivo del juego es realizar la combinación de un trío y una escalera con las cartas dadas. Un trío equivale a tres cartas del mismo valor y una escalera equivale a cuatro cartas consecutivas. Los tríos y las escaleras pueden estar compuestos por cartas de diferentes palos. En cada turno el jugador debe escoger una carta; esta puede ser la última carta jugada por el oponente o una nueva carta de la baraja. El jugador puede acomodar las cartas para formar el trío y la escalera y debe descartar una carta al final del turno (que será la carta disponible para el siguiente turno del oponente). Una vez terminada su jugada el jugador puede terminar el turno (lo cual habilita el turno del oponente) o cantar victoria. Cuando se canta una victoria, el servidor determina si es válida o no y le informa a los jugadores. En caso de que sea una victoria inválida, gana el oponente.

El programa distribuido consta de dos partes: un programa servidor, encargado de mantener la información estadística del juego y de permitir a los jugadores encontrarse para una partida; y un programa cliente, a través del cual un usuario puede jugar “Continental”.

El programa cliente debe ofrecer las siguientes funcionalidades:

1. Conectarse al servidor. Para esto el usuario debe suministrar su nombre, la dirección IP del servidor al cual se quiere conectar y el puerto por el cual dicho servidor se encuentra esperando conexiones.
2. Seleccionar una nueva carta de la baraja o la última carta jugada por el oponente.
3. Terminar su turno, en cuyo caso descarta la carta sobrante y la deja disponible para su oponente.
4. Cantar victoria.
5. Mostrar su estado (esperando un oponente, realizando jugada, esperando jugada y juego terminado).
6. Mostrar las estadísticas de su oponente: nombre, cantidad de victorias y derrotas.

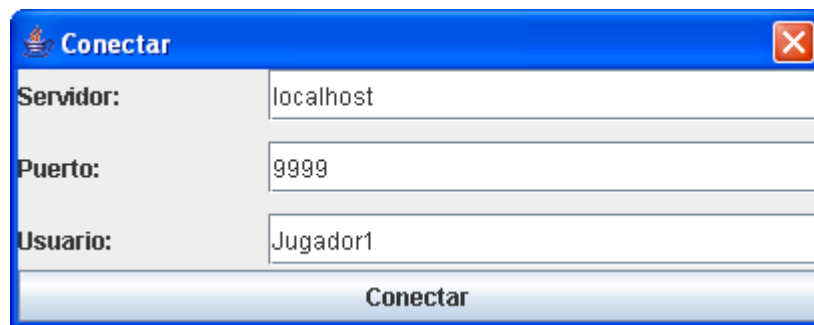
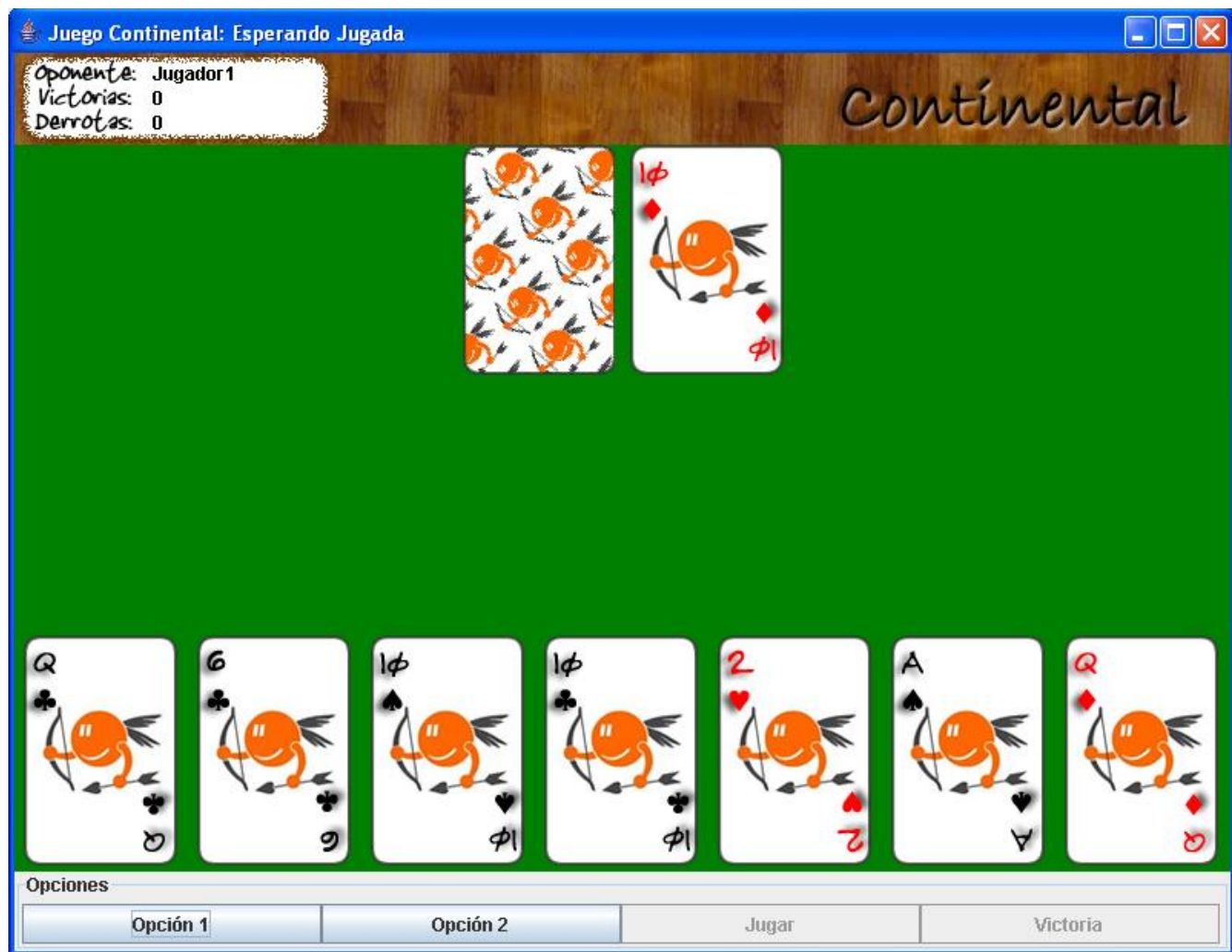
El programa servidor, por su lado, debe esperar a que los jugadores se vayan conectando y, por cada pareja que pueda armar, inicia un juego. Además, el servidor debe ofrecer las siguientes opciones al usuario:

1. Mostrar los juegos que se encuentren en curso.
2. Mostrar las estadísticas históricas de los jugadores. Se debe mostrar el nombre de cada jugador que haya participado en un juego, y la cantidad de victorias y de derrotas.

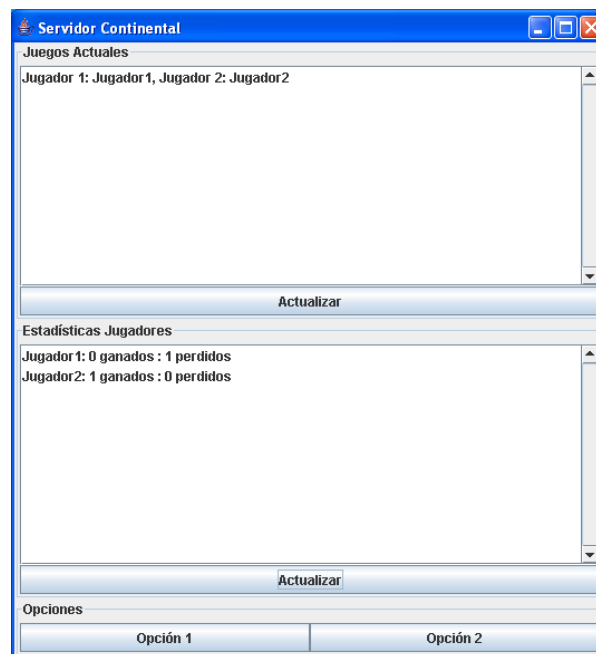
La información histórica de los juegos debe ser persistente, de manera que cada vez que se ejecute de nuevo el servidor, las estadísticas sean presentadas al usuario.



## Interfaz Cliente



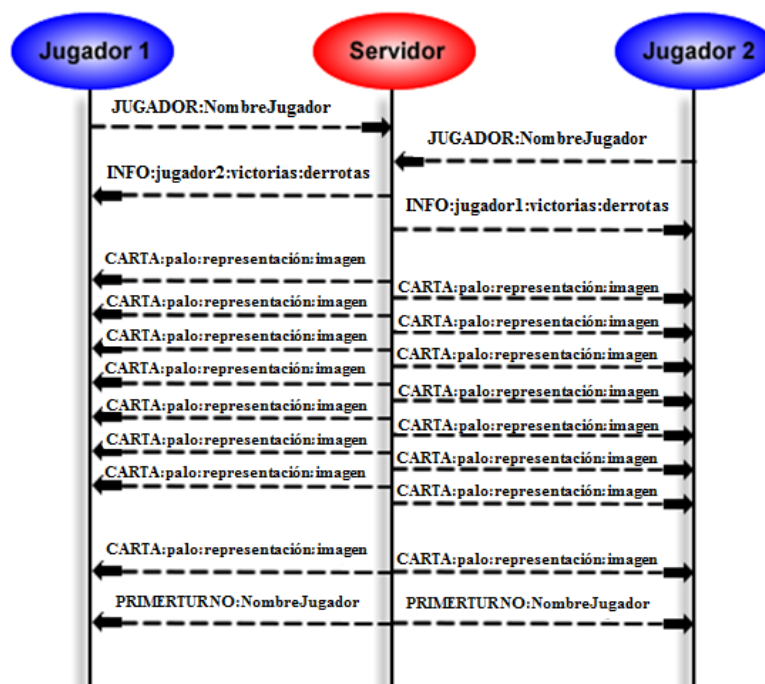
## Interfaz Servidor



## Protocolo de Comunicación

A continuación se presenta el protocolo de comunicación que establece cuáles mensajes (y en qué orden) se deben enviar para realizar cada una de las tareas del sistema. Es responsabilidad tanto del servidor como de los clientes ser capaz de interpretar este protocolo y realizar las tareas necesarias.

### Conexión de un jugador

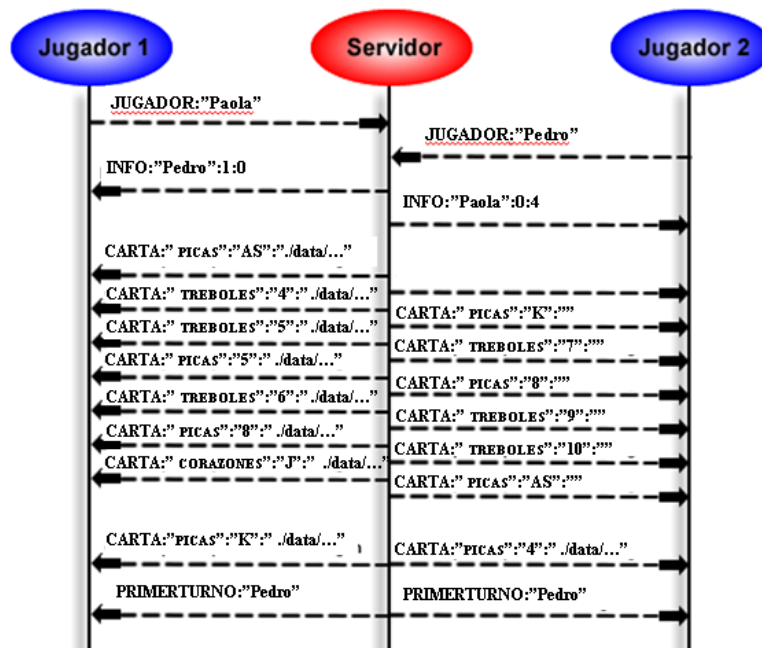


El primer jugador envía un mensaje al servidor indicando su nombre (mensaje: JUGADOR). El servidor espera a que se conecte otro usuario para poder crear el juego. Una vez se conecta el segundo jugador (informando su nombre con el mensaje JUGADOR), el servidor envía las estadísticas de cada jugador a su respectivo oponente (mensaje: INFO). Posteriormente, el servidor envía siete cartas a cada jugador (mensajes: CARTA). Por último, el servidor envía a cada jugador la primera carta destapada de la baraja (mensaje: CARTA) e informa (mediante un mensaje PRIMERTURNO) a los dos jugadores, quién tiene el primer turno.

La sintaxis de cada mensaje es explicada en la siguiente tabla:

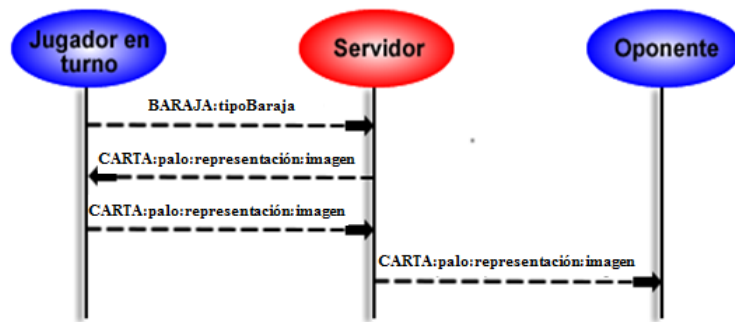
Mensaje	Parámetros	Descripción
JUGADOR	nombreJugador	Representa el nombre del jugador que se va a conectar al servidor
INFO	jugador	Representa el nombre del oponente
	victorias	Representa el número de victorias del oponente
	derrotas	Representa el número de derrotas del oponente
CARTA	palo	Representa el palo de la carta (picas, corazones, diamantes o tréboles)
	representación	Es el número o letra de la carta
	imagen	Es la ruta de la imagen asociada a la carta
PRIMERTURNO	NombreJugador	Representa el nombre del jugador que se empieza el juego

A continuación un ejemplo concreto:



Para este ejemplo, el primer jugador envía el mensaje al servidor indicando su nombre: "Paola". En este punto, el servidor espera a que se conecte otro usuario para poder crear el juego. Luego, el segundo jugador se conecta informando su nombre: "Pedro". En ese momento, el servidor envía las estadísticas de Pedro a Paola y las estadísticas de Paola a Pedro. Posteriormente, el servidor envía siete cartas a Paola y siete cartas a Pedro. Adicionalmente, el servidor envía tanto a Paola como Pedro su primera carta destapada de la baraja, a Paola se le entrega la K de PICAS y a Pedro el 4 de PICAS. Por último, el servidor informa a los dos jugadores que Pedro tendrá el primer turno.

## Realizar una jugada

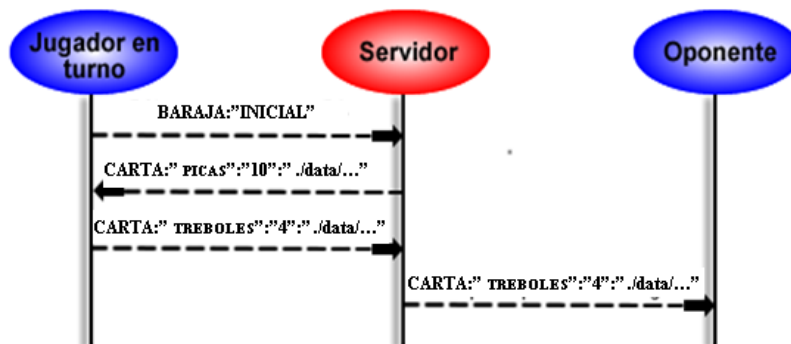


El jugador en turno envía un mensaje al servidor indicando de cuál baraja desea tomar una carta (mensaje: BARAJA). La baraja puede ser la inicial o la jugada; la baraja inicial es la baraja que se encuentra tapada y no se sabe qué carta se va a despachar; la baraja jugada es la baraja en la cual se ubican las cartas que han sido descartadas por los jugadores en cada turno. Una vez enviado el mensaje con la información de la baraja, el servidor le envía al jugador un mensaje con información de la carta correspondiente (mensaje: CARTA), con la cual el jugador podrá realizar su juego y posteriormente descartar una carta para quedar con un total de siete cartas de nuevo. El jugador informa al servidor cuál es la carta descartada (mensaje: CARTA) y el servidor a su vez informa al oponente cuál es esta carta (mensaje: CARTA) para que actualice su información.

La sintaxis de cada mensaje es explicada en la siguiente tabla:

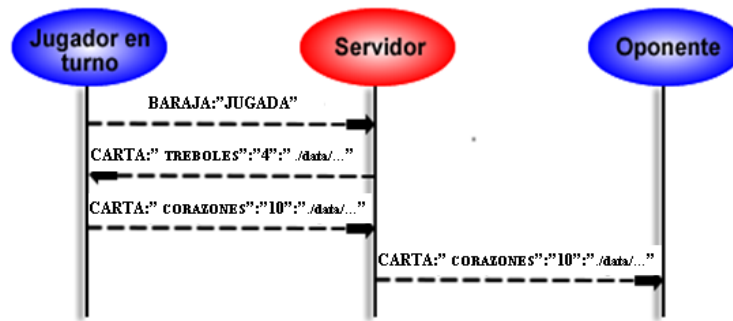
Mensaje	Parámetros	Descripción
BARAJA	tipoBaraja	Representa el tipo de baraja de la cual el jugador desea tomar la carta para jugar. Puede tomar dos valores: INICIAL (para indicar que desea la carta de la baraja tapada) o JUGADA (para indicar que desea tomar la última carta jugada y descartada por el oponente)
CARTA	palo	Representa el palo de la carta (picas, corazones, diamantes o tréboles)
	representación	Es el número o letra de la carta
	imagen	Es la ruta de la imagen asociada a la carta

A continuación un ejemplo concreto cuando se toma una carta de la baraja inicial:



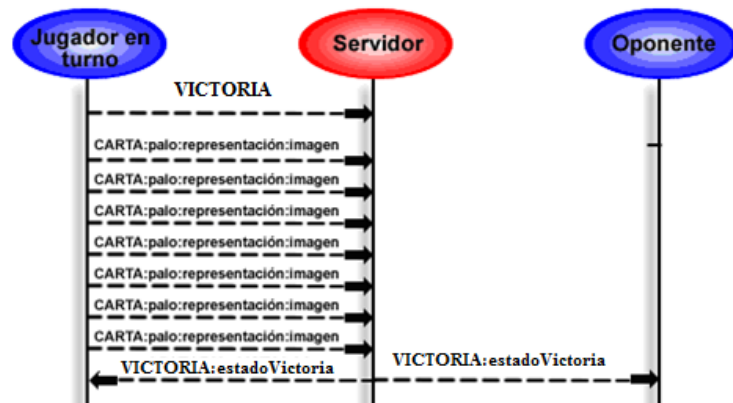
En este ejemplo, el jugador en turno envía un mensaje al servidor indicando que desea tomar una carta de la baraja inicial, es decir, una carta de la baraja que se encuentra tapada. Luego, el servidor le envía al jugador en turno una carta de esta baraja, en este caso le envía el 10 de PICAS. Posteriormente, este jugador podrá realizar su juego y cuando termine su turno, envía al servidor la carta que descarta: 4 de TREBOLES. Finalmente, el servidor envía este 4 de TREBOLES al oponente (esta carta quedará disponible en su respectiva baraja jugada).

A continuación un ejemplo concreto cuando se toma una carta de la baraja jugada:



En este ejemplo, el jugador en turno envía un mensaje al servidor indicando que desea tomar una carta de la baraja jugada, es decir, la carta que descartó el oponente en su turno anterior. Luego, el servidor le envía al jugador en turno la primera carta de esta baraja, en este caso le envía el 4 de TREBOLES. Posteriormente, este jugador podrá realizar su juego y cuando termine su turno, envía al servidor la carta que descarta: 10 de CORAZONES. Finalmente, el servidor envía este 10 de CORAZONES al jugador oponente (esta carta quedará disponible en su respectiva baraja jugada).

## Declararse ganador o cantar victoria



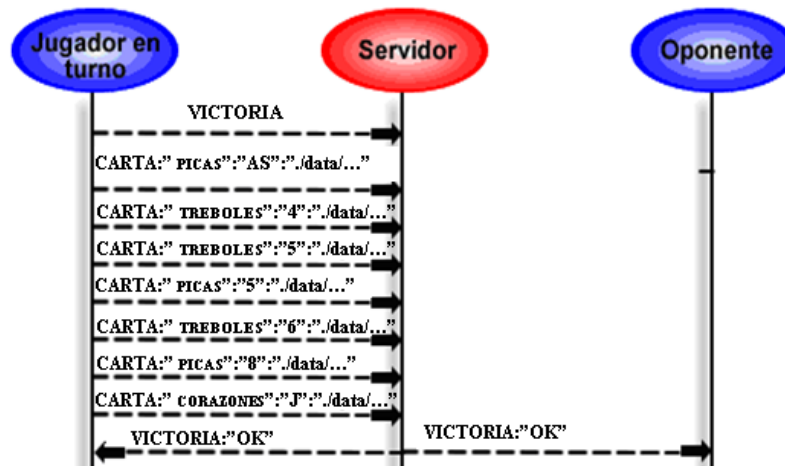
Cuando un jugador cree que tiene una jugada correcta, este puede cantar victoria al servidor (mensaje: VICTORIA). Inmediatamente después el jugador envía al servidor el juego para que el servidor valide la victoria (mensajes: CARTA). El servidor revisa si la victoria es válida o no y envía un mensaje al jugador y al oponente indicando si la victoria es válida (mensaje: VICTORIA).

La sintaxis de cada mensaje es explicada en la siguiente tabla:

Mensaje	Parámetros	Descripción
VICTORIA	N/A	Representa el mensaje que envía un jugador cuando quiere declarar la victoria
VICTORIA	estadoVictoria	Representa la respuesta del servidor determinando el estado de la victoria. Puede tomar dos valores: OK (para indicar que la victoria es válida) o FALSA (para indicar que la victoria es inválida)
CARTA	palo	Representa el palo de la carta (picas, corazones, diamantes o tréboles)
	representación	Es el número o letra de la carta
	imagen	Es la ruta de la imagen asociada a la carta



A continuación un ejemplo concreto:



En este ejemplo, el jugador en turno cree tener la victoria y la canta: informa al servidor su posible victoria y le envía las siete cartas de su juego (AS de PICAS, 4 de TREBOLES, 5 de TREBOLES, 5 de PICAS, 6 de TREBOLES, 8 de PICAS y J de CORAZONES). Posteriormente, el servidor revisa si la victoria es válida o no y envía un mensaje al jugador y al oponente indicando que la victoria es válida.

## Persistencia

Se utilizará como motor de base de datos el sistema (*open source*) llamado Derby, desarrollado en el proyecto Apache Derby.

Las ventajas de utilizar un motor de base de datos Derby son:

- Está desarrollado completamente en Java y tiene un sistema de persistencia basado en archivos.
- Permite crear un servidor Derby independiente de la aplicación y establecer la comunicación con esta a través de sockets y desde diferentes máquinas.
- Es una base de datos embebida, es decir, que se puede utilizar como si fuera parte de la aplicación.

Para la aplicación se usará la base de datos embebida que consta de una única tabla para almacenar los resultados de los jugadores desde la primera vez que el servidor es ejecutado. Los campos de esta tabla son: nombre del jugador, victorias y derrotas. La llave primaria de la tabla es el nombre del jugador. Esto implica que no puede haber dos registros que tengan el mismo valor en su llave primaria (no puede haber en la tabla dos jugadores con el mismo nombre). El siguiente cuadro muestra la tabla de la base de datos con sus campos y con su llave primaria.

Tabla	Campos	Llave primaria
resultados	nombre, victorias, derrotas	nombre