Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Computación e Informática

Desarrollo de Aplicaciones para Internet
CI-2413
Grupo 1
Braulio Solano

Tarea Programada #2

Estudiantes
Xiannie Rivas Hylton – B05206
Jenny Vásquez Calderón – B17016

Sede Rodrigo Facio, Martes 23 de Junio de 2015

Tabla de Contenidos

Introducción

Descripción del problema

Esta tarea consiste en la implementación del juego Ahorcado o Gato, que por medio de Servicios Web, permite que el juego ejecute la lógica programada en PHP en un cliente cuya interfaz puede ser programada en diversos lenguajes de programación (en este caso .NET).

Además se debe guardar en una base de datos un registro de los mejores puntajes ordenados por tiempo de duración de sus juegos en segundos.

Para realizar esto se implementó una clase en PHP que se encarga de la lógica del juego y un cliente SOAP cuya interfaz cuya interfaz fue programada en .NET utilizando el IDE *Visual Studio*.

Descripción de la metodología y análisis del problema

Para resolver este problema, se requirió de la implementación de una clase en PHP llamada Ahorcado.class.php, que posee los métodos y variables necesarias para la correcta ejecución del juego. A continuación se describen brevemente dichos métodos:

- getTiempo(). Retorna el tiempo que le toma al jugador descubrir la palabra correcta.
- getPalabraOculta(). Retorna la palabra escogida aleatoriamente del banco de palabras con sus caracteres intercambiados por un "_".
- juego($letra). Se encarga de ejecutar la dinámica del juego. Recibe la letra digitada por el usuario, se verifica que esta se encuentre en la palabra. De ser cierto lo anterior, se intercambia la posición o posiciones en la palabra oculta cuyas letras corresponden a la letra digitada y retorna la nueva palabra oculta.
- verificarJuego(). Verifica que el jugador aún tenga intentos.

- guardarPuntaje(). En caso de que el jugador pueda adivinar la palabra, su puntaje se guarda en la base de datos.
- getMejores(). Retorna los mejores 10 puntajes.

Por otro lado la implementación de la interfaz, fue realizada en .NET. Cada vez que el usuario comete un error se le muestra una imagen que indica su estado de "ahorcado", como se realizaría tradicionalmente. Además, por medio del botón "Puntajes" se despliega una ventana que muestra los 10 mejores puntajes (por medio de un llamado al método getMejores() de la clase Ahorcado.class.php).

Diseño de clases UML

La Figura 1 muestra el diseño de clases UML generado por medio de *Visual Studio*:

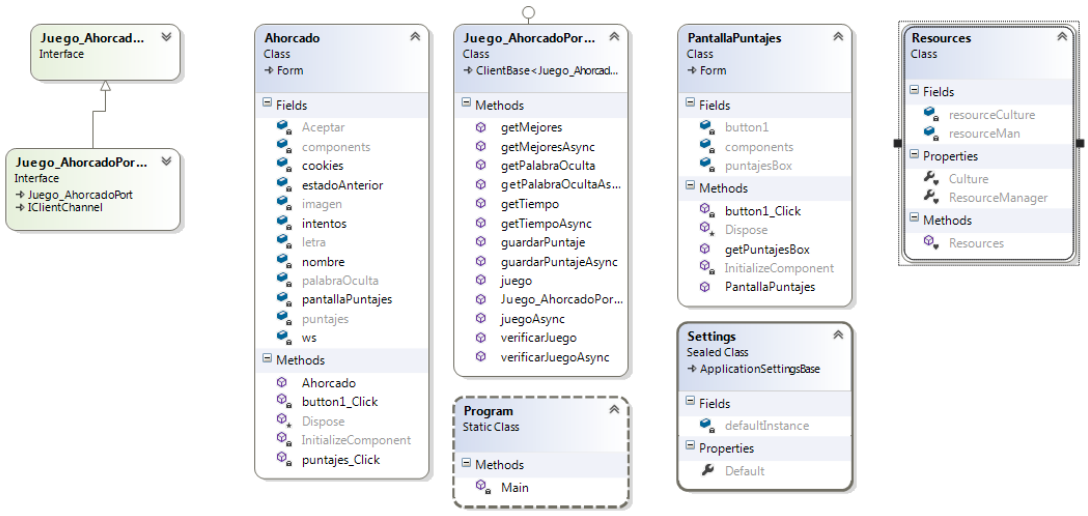

Figura 1. Diagrama de clases UML.

Casos de prueba y análisis de resultados

En esta sección se mostrarán unos cuantos casos de uso de la aplicación en ejecución.

Usuario no adivina la palabra

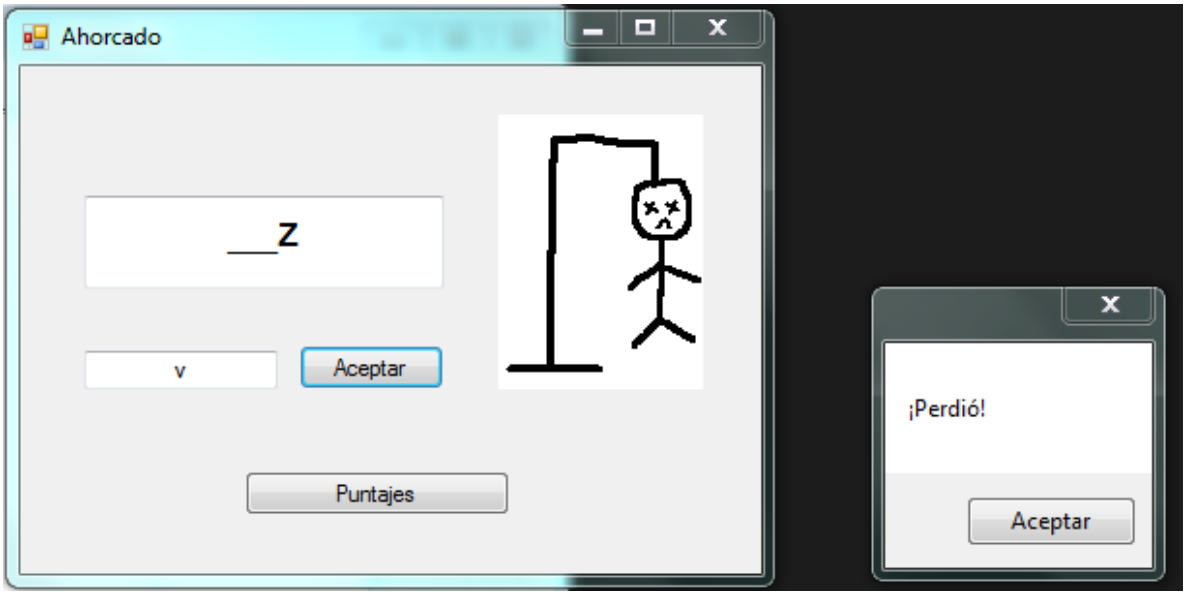

Figura 2. Usuario no logra adivinar la palabra oculta.

Como se muestra en la figura anterior, en el caso en el que el usuario no logra adivinar la palabra, se le va mostrando progresivamente el dibujo del ahorcado con cada fallo hasta llegar al punto de la Figura 2. Adicionalmente se le despliega una alerta que indica en su mensaje claramente que perdió la partida. El puntaje de un usuario que pierde no se guarda en la base de datos.

Usuario adivina la palabra

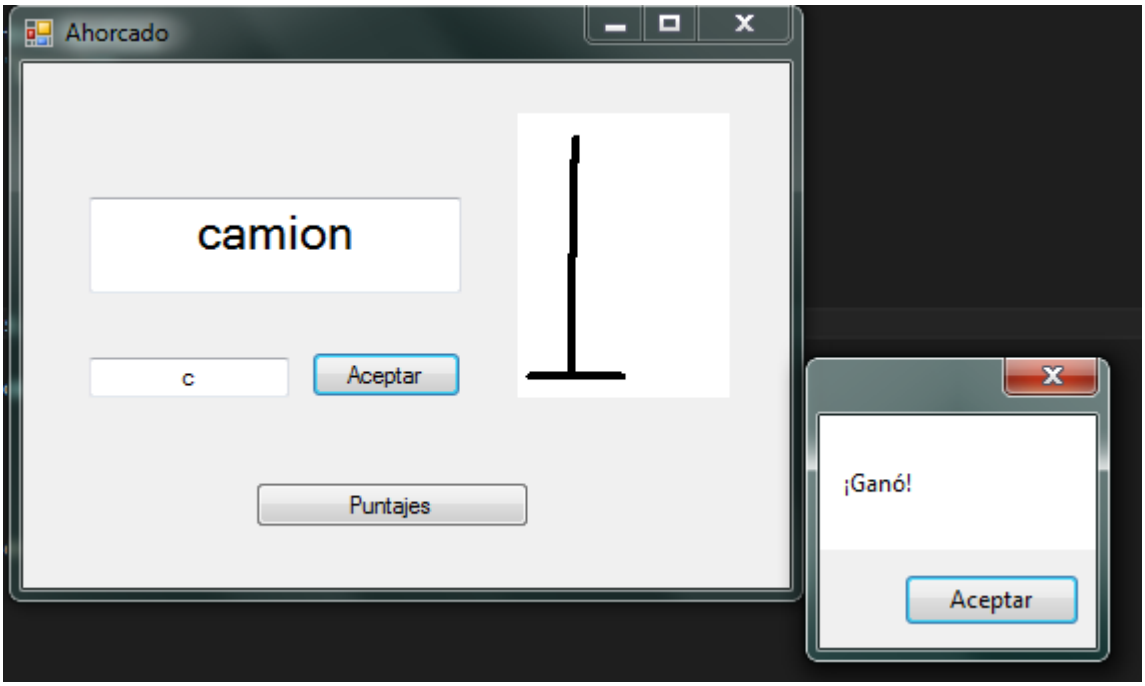

Figura 3. Usuario logra adivinar la palabra.

Como se muestra en la Figura 3, una vez que el usuario logra adivinar la palabra correctamente se le despliega un mensaje que le indica que ganó. Además, su puntaje es guardado y es posible observarlo (si se encuentra dentro de los 10 mejores) al presionar el botón de Puntajes como se muestra en la Figura 4.

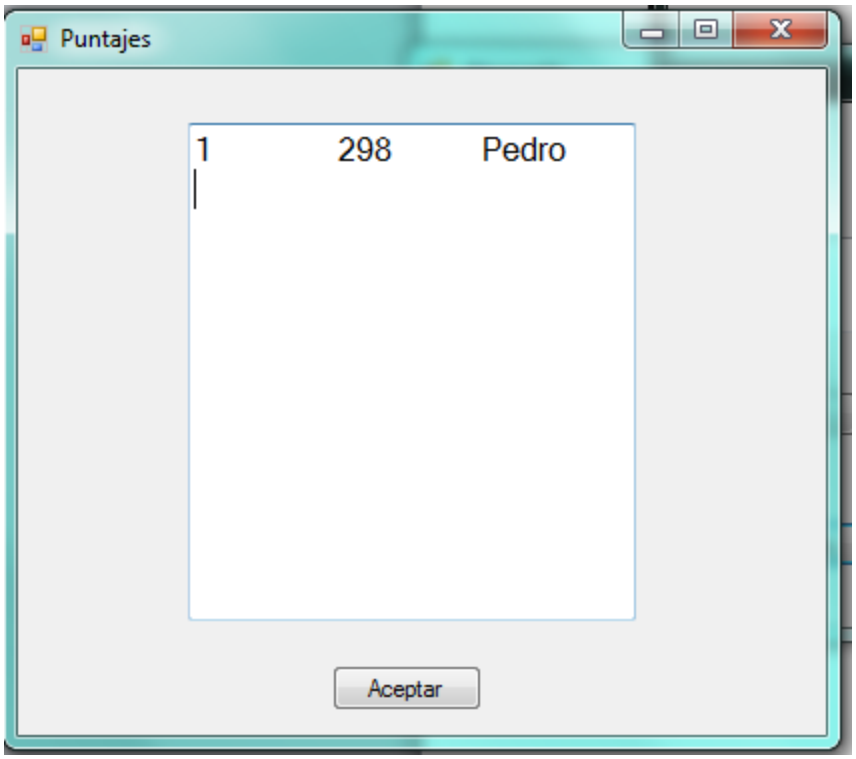

Figura 4. Mejores puntajes.