INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS MONTERREY



Desarrollo de Aplicaciones Web

Proyecto final

“Jeonatra”

Azael Alberto Alanis Garza - A01175470

José Luis Loya Cabrera – A01111663

Iván Díaz Martínez – A01175445

Índice

Introducción: 3

Descripción de Jeopardy: 3

Descripción de la tecnología: 3

Arquitectura del servidor: 4

Clases y diagrama entidad-relación: 5

Manual de instalación: 6

Manual del usuario: 6

Referencias externas: 6

# Introducción:

Para el proyecto final de la clase de Desarrollo de Aplicaciones Web (DAW) nos adentramos en la tarea de crear una aplicación inspirada en el famoso juego estadounidense Jeopardy, para el desarrollo de la aplicación decidimos utilizar tecnologías ‘back-end’ como Ruby y Sinatra junto con tecnología ‘front-end’ como JavaScript, jQuery, CSS, HTML, Ajax y Less para hacer un sitio web interactivo y atractivo visualmente.

# Descripción de Jeopardy:

Jeopardy es un concurso de televisión estadounidense creado por Merv Griffin. El concurso cuenta con preguntas tipo trivia que abarcan numerosos temas entre los que se incluye historia, idiomas, literatura, cultura popular, bellas artes, ciencias, geografía, y deportes. Su mecánica consiste en que uno de los tres concursantes elige uno de los paneles del tablero de juego, el cual, al ser descubierto, revela una pista en forma de respuesta; los concursantes deben dar sus respuestas en forma de una pregunta.

# Descripción de la tecnología:

Ruby: es un lenguaje de programación interpretado, reflexivo y orientado a objetos. Ruby sigue el "principio de la menor sorpresa", lo que significa que el lenguaje debe comportarse de tal manera que minimice la confusión de los usuarios experimentados. Ruby es orientado a objetos: todos los tipos de datos son un objeto, incluidas las clases y tipos que otros lenguajes definen como primitivas, (como enteros, booleanos, y "nil"). Toda función es un método.

Sinatra: es un framework para aplicaciones web de software libre y código abierto, y lenguaje específico del dominio escrita en Ruby. Diseñado y desarrollado por Blake Mizerany, Sinatra es pequeño y flexible. Sinatra no sigue el típico patrón modelo vista controlador que se ve en otros frameworks, como Ruby on Rails. En su lugar, Sinatra se enfoca en la "rápida creación de aplicaciones web en Ruby con el mínimo esfuerzo."

JavaScript: es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor

jQuery: es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

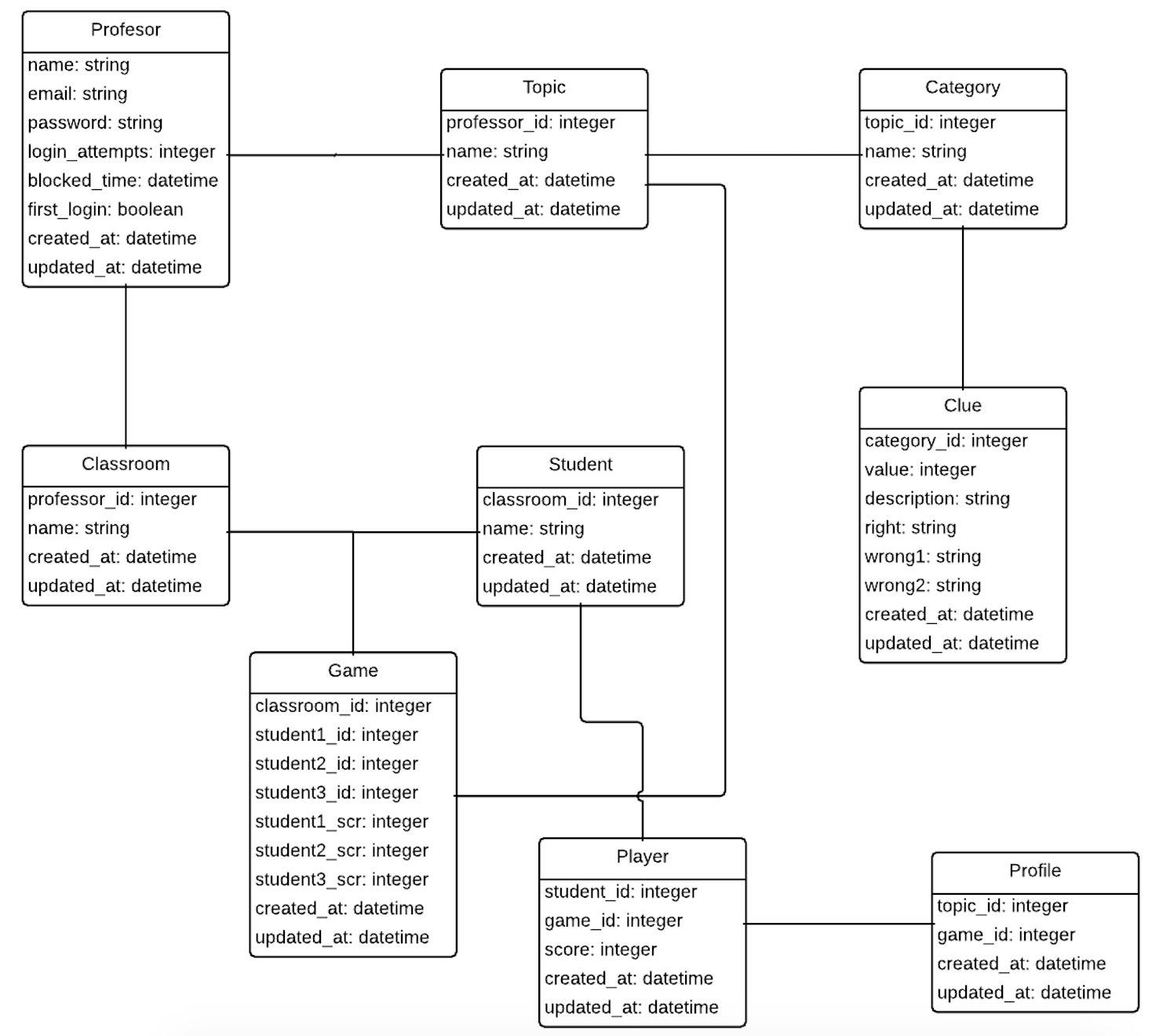
Ajax: Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

# Arquitectura del servidor:

La arquitectura que decidimos implementar es cliente servidor, que es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes'', resultan en un trabajo realizado por otros computadores llamados servidores".

El sitio donde decidimos hacer el “hosting” de nuestra aplicación es Heroku.

# Clases y diagrama entidad-relación:



El diagrama de clases presentado anteriormente muestra los 9 modelos que utilizamos en toda la aplicación, los atributos se muestran en la imagen junto con sus relaciones.

* Profesor
* Tema
* Categoría
* Clase
* Estudiante
* Pista
* Juego
* Jugador
* Perfil

# Manual de instalación:

Hay 2 maneras de correr nuestra aplicación, la más sencilla es por medio del sitio web de Heroku donde está siendo ‘hosteada’ y sencillamente utilizarla desde ahí, ya tiene configuradas las bases de datos, las relaciones y todo lo necesario para poder ejecutar completamente la aplicación.

Por otro lado, también se puede instalar de manera local en cualquier computadora pero es un proceso más tardado, a continuación se presentan los pasos necesarios:

1. Clonar el repositorio de git a cualquier lugar del disco duro

$ git clone <https://github.com/azaelalanis/Jeonatra.git>

1. Estando en el directorio del repositorio tienes que correr el comando que sirve para instalar todas las dependencias que necesita la aplicación para funcionar de manera correcta, estas dependencias están incluidas en un archivo llamado Gemfile:

$ bundle install

1. Posteriormente es necesario migrar la base de datos para crearla y a las tablas también.

$ rake db:migrate

1. Habiendo hecho todo esto sencillamente puedes ingresar a la aplicación dirigiéndote a terminar y corriendo el comando:

$ rackup

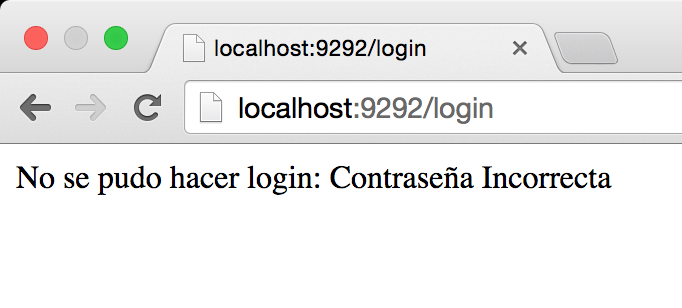
1. Esto pondrá a correr la aplicación en algún puerto que te indicara la terminal por el cual podrás acceder directamente desde *localhost:9292*
2. A partir de aquí puedes utilizar la aplicación de manera local en tu computadora.

\*Se espera que el usuario a instalar Jeonatra tenga instalados Ruby, Sinatra, Postgres y cualquier otro repositorio necesario para ejecutar la aplicación, ya que instalar estas cosas no se cubren en el tutorial.

# Manual del usuario:



Pantalla de inicio de sesión, donde se ingresa el correo del usuario y contraseña.



Si el usuario se equivoca al ingresar la contraseña le mostrara un mensaje, si se equivoca más de 3 veces, se bloqueará la cuenta por un tiempo.



Tendrás que esperar a que ese tiempo se agote para que se desbloquee la cuenta y puedas ingresar nuevamente al sistema.



Esta es la pantalla donde se registra un nuevo usuario, se escribe correo y contraseña y al momento de registrar se manda un correo automáticamente con una contraseña temporal que tendrá que cambiar al momento de entrar al sistema por primera vez.

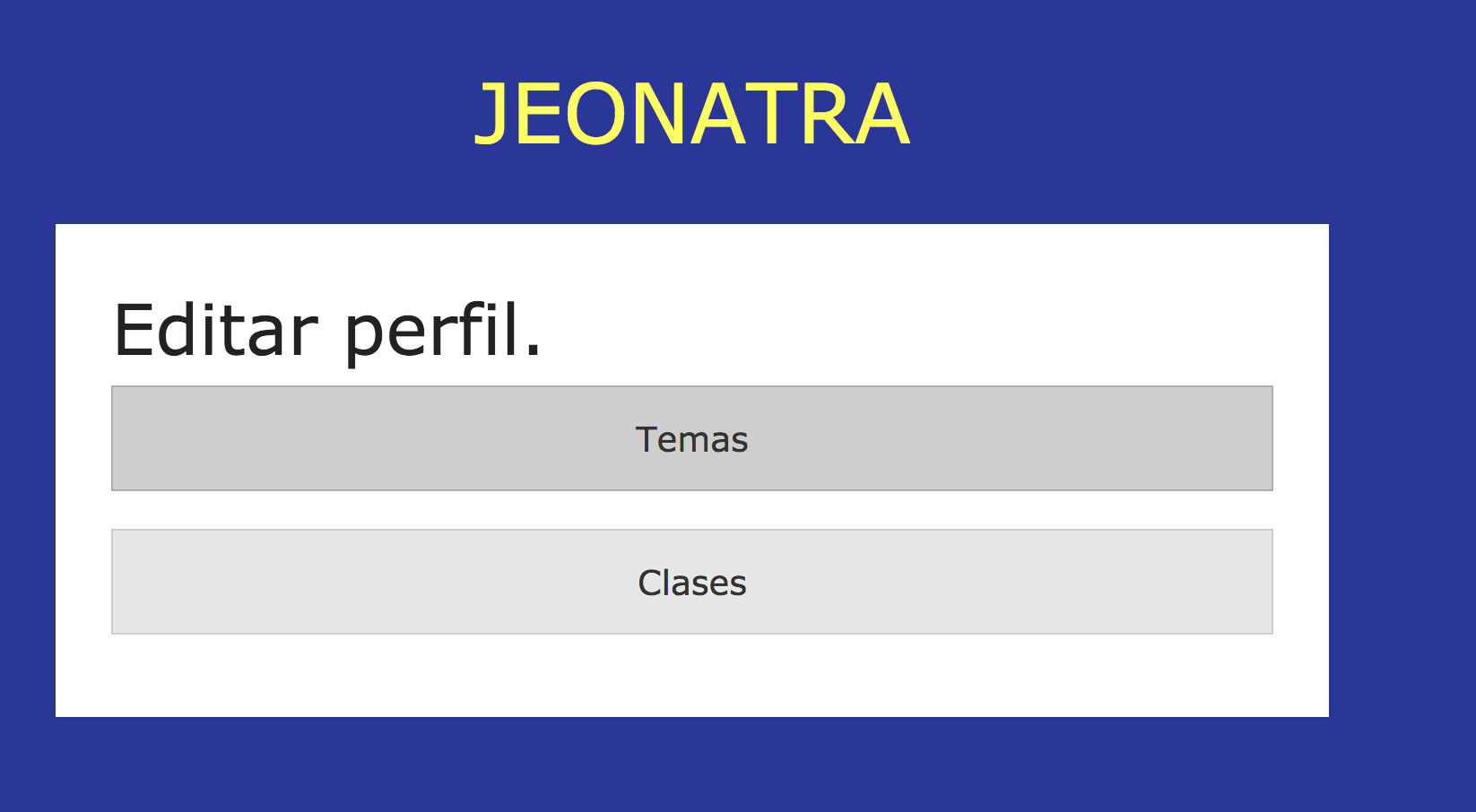


Una vez que cambias la contraseña siendo un usuario nuevo puedes ingresar al sistema normalmente.

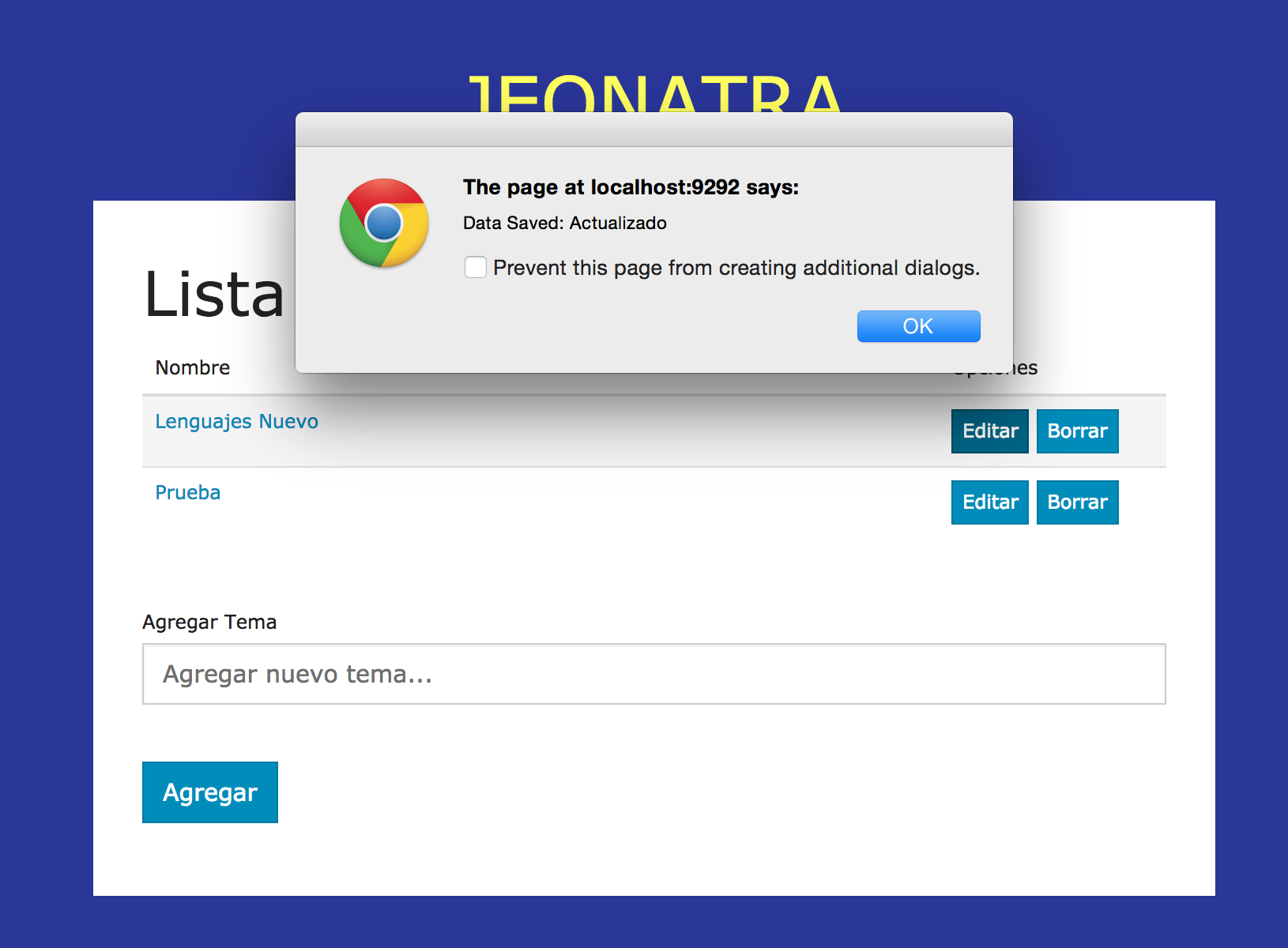


Aparecen 3 opciones, la primera es para crear un nuevo juego, la segunda para editar el perfil del profesor (en donde se pueden editar los temas, categorías, pistas de cada categoría, así como las clases del profesor y los alumnos de cada clase.

Esto se muestra en las siguientes pantallas.

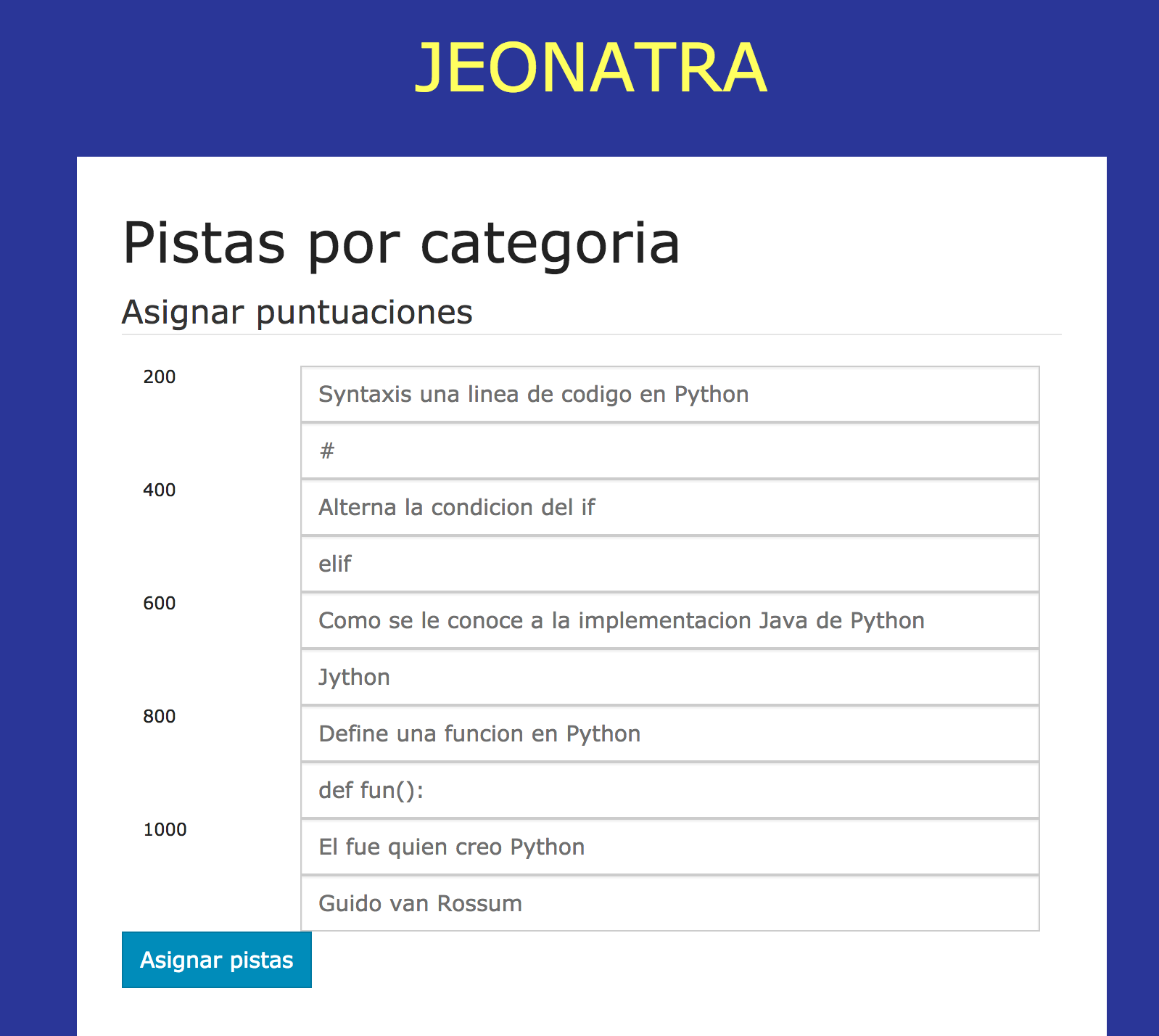




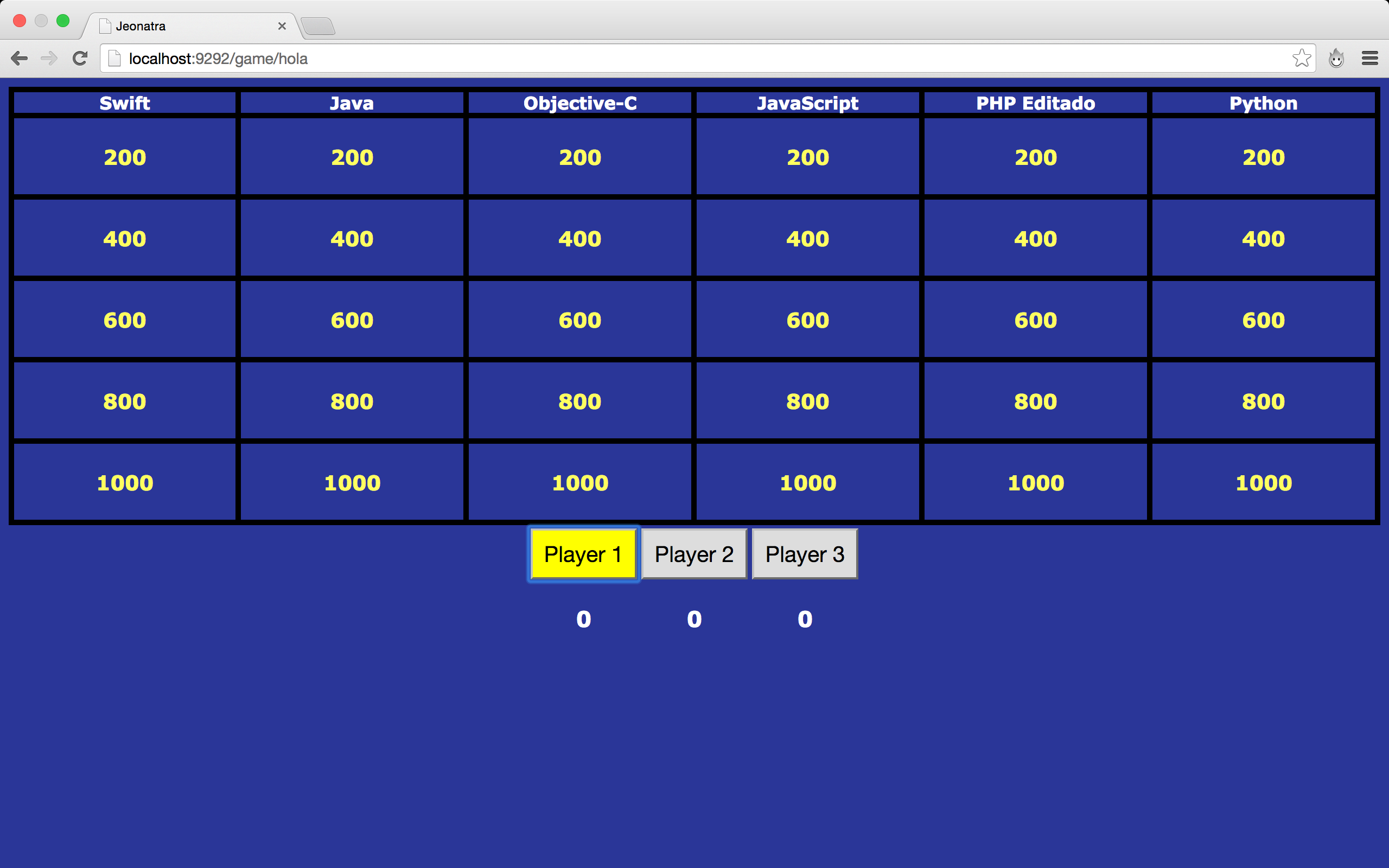


El editar cada campo se actualiza automáticamente por medio de Ajax en la base de datos.





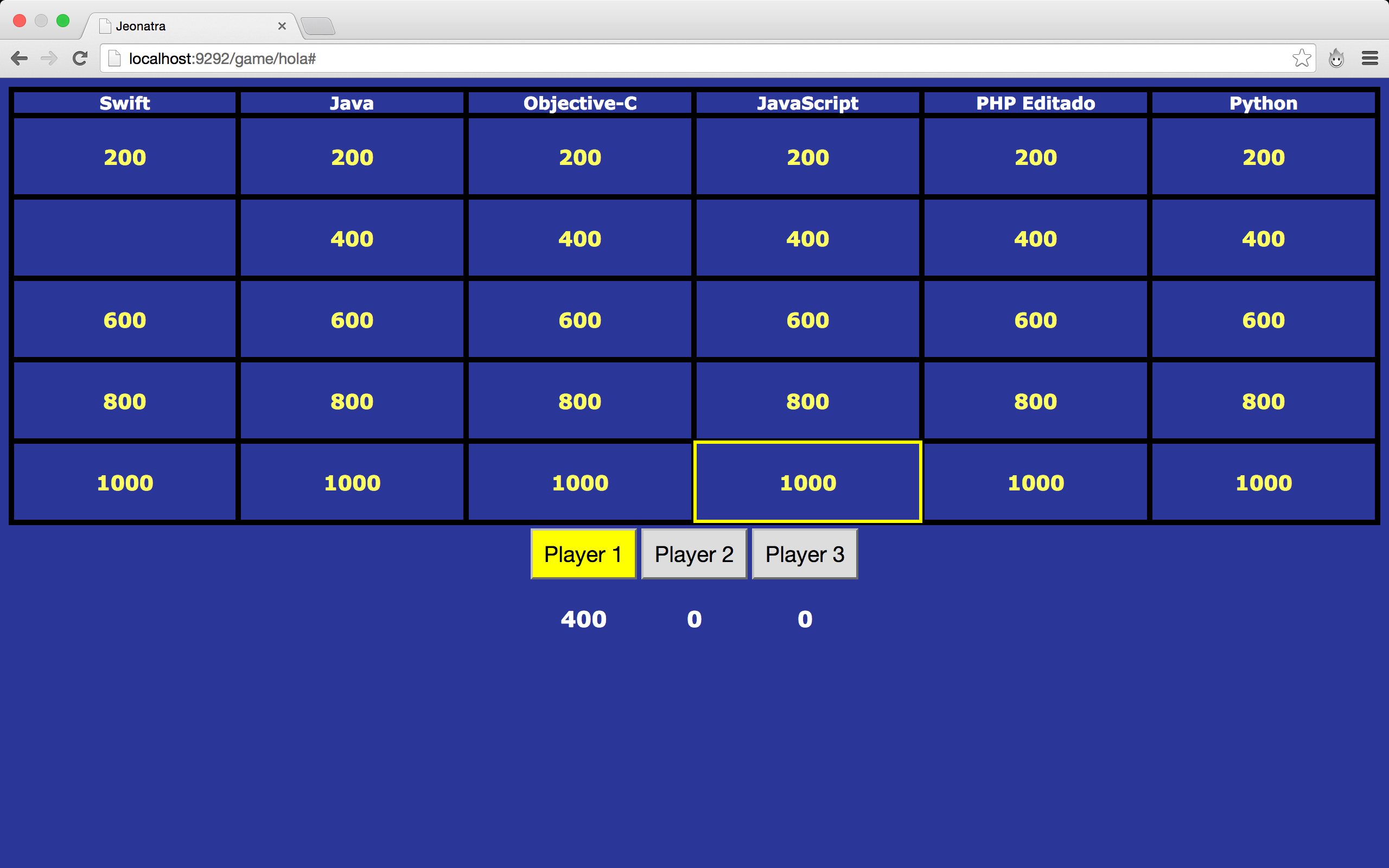
Por ultimo cada categoría muestra las preguntas que tiene y en esta pantalla te permite editarlas para que automáticamente se actualicen en la base de datos y puedas jugar con la información más nueva.

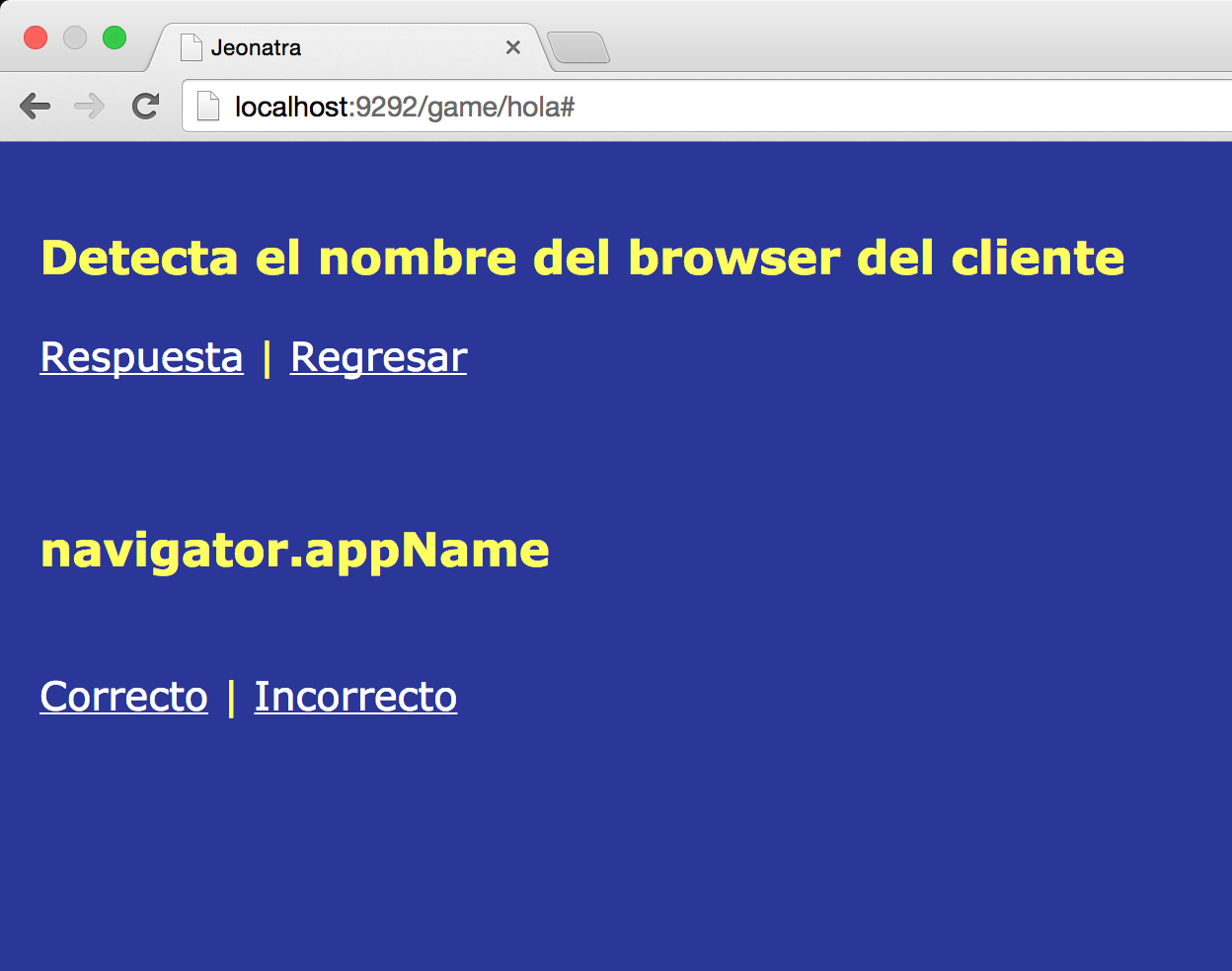


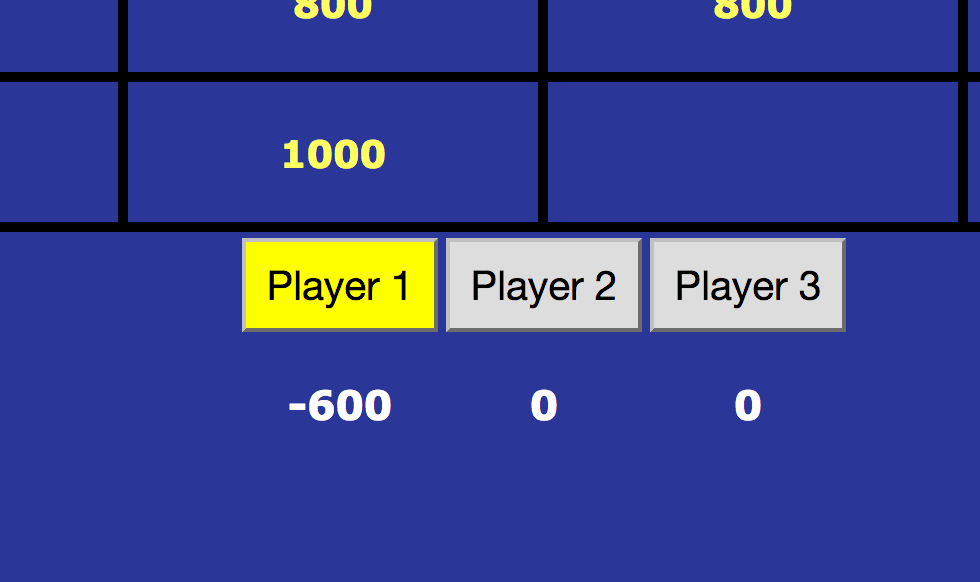
Al estar jugando se muestran los tres alumnos y se denota el jugar activo con color amarillo.



Al presionar una pregunta te aparece la pregunta y el alumno dice la pregunta, el profesor puede mostrar la pregunta y en base a su criterio determinar si la pregunta fue correcta o incorrecta, en caso de ser correcta los puntos se suman al usuario y si no, se le restan de la misma manera.







# Referencias externas:

Repositorio de GitHub: https://github.com/azaelalanis/Jeonatra