



UNIVERSITAT DE
BARCELONA

Pràctica 2. Creación de una página web con Flask/Vue. REST-API.

Assignatura: Software Distribuït

Gerard Perelló
Ignacio Morera

A lo largo de 7 sesiones, cada una correspondiente a una semana de trabajo, hemos creado una página web funcional, deployada utilizando la plataforma en la nube de *Heroku*.

Comenzamos creando un usuario. Al utilizar la base de datos de *Heroku*, podemos detectar cuando dos usuarios coinciden y denegar la petición. También sabremos si posteriormente, al hacer el login, te has equivocado de contraseña. Hemos creado dos tipos de usuario a disposición del corrector de la práctica: uno de tipo *user*, con el usuario *1* y contraseña *1*; y otro de tipo *admin*, con el usuario *admin* y la contraseña *admin*.

Hasta hace un par de días teníamos diversos problemas configurando la base de datos al realizar el deploy y no se realizaba bien la verificación de cuentas. Recibimos muchas quejas de que fuera cual fuera el nombre de usuario saltaba el error de que dicho usuario ya existía. Gracias a estas quejas conseguimos encontrar el problema y arreglarlo. Lo mismo sucedía al loguearse, ya que habíamos introducido a mano diferentes usuarios, pero no los reconocía. Verificamos que el problema venía de la función encargada de verificar que la contraseña fuera correcta. Como no encontrábamos una solución al respecto, guardábamos las contraseñas sin *hashear*. Es una solución que utilizamos como último recurso, a pesar de que habíamos conseguido implementar la función en local. Al final conseguimos solucionarlo. El problema era que el tamaño del hash era demasiado grande, nuestra base de datos no podía guardarlo. La solución fue aumentar el tamaño de la base de datos.

Una vez *logueados*, encontramos un menú personalizado para el usuario. Se realizará un *get* a base de datos de todos los *matches* disponibles. El único límite a la compra lo dará el dinero disponible, personal para cada usuario y que obtendremos junto con el *get* de *matches*. Dependiendo del dinero del que dispongamos nos permitirá comprar una cierta cantidad de entradas para un determinado partido. Si no nos alcanza el dinero el botón de *Añadir a la cesta* se bloqueará. Estas compras son posibles únicamente siendo un usuario de tipo *user*. Sin embargo, como nuestros compañeros no pudieron crear usuarios en la sesión de *testing* no hemos recibido *feedback* al respecto. Hemos visto muchos comentarios de que no se podía ver la cesta de la compra. También han habido otros que destacaban la falta de bugs (probablemente debido a que se había reducido el número de funcionalidades) y que al menos se permitía navegar entre las ventanas de *matches*, *login* y *register*.

Una vez hemos realizado las compras pertinentes (para las cuales podemos revisar la cantidad de saldo que nos queda en el icono de dinero arriba a la derecha) podemos acceder a la vista de la cesta de la compra. En esta vista veremos los *matches* con la cantidad de entradas correspondiente para cada partido. Cabe destacar que en nuestro caso no se presentan *bugs* como tener una entrada para cada orden de compra y no para cada *match*, como algunos de nuestros compañeros. También hemos truncado el número de decimales que muestra el precio de cada *match*, como nos hizo notar un compañero (antes de la sesión de *testing*).

Una vez hayamos acabado de seleccionar partidos podemos darle al botón de compra para ejecutar las órdenes. El dinero se descontará inmediatamente en la página web y en la base de datos, así como el número de *matches* disponibles y las órdenes para el usuario. De esta manera, veremos de forma inmediata las consecuencias de la compra, tanto si recargamos la página como si cambiamos de vista.

En cuanto al usuario de tipo *admin*. Hemos comentado que existen un par de diferencias respecto al usuario anterior. En este caso no podremos comprar entradas ni acceder a la cesta de la compra. Ambas opciones aparecerán deshabilitadas. Sin embargo, aparecerá un nuevo botón para cada partido, que nos permitirá eliminarlo tanto para la página web como en el *backend*. También podremos realizar un formulario para crear un nuevo *match*. En este caso, al ser un usuario *admin* hemos pensado que tendrá acceso a la base de datos, por lo que será más cómodo crear un partido escribiendo los *id* de los equipos y de la competición, en lugar de poner un nombre que coincida exactamente con el de dicho equipo o competición.

Finalmente, cabe destacar que la URL de la página funcional de Heroku es: (con el 1 delante)

<https://a12-sportsmaster1.herokuapp.com/>