

Documentación de la página web del proyecto Smart Parking

Documentación realizada por Marco Novaes (marco.novaesm@alumnos.upm.es)

Proyecto accesible desde <https://github.com/marco-novaes98/smartParking>

1. Habilitar una Raspberry PI como servidor

En caso de poder recuperar la misma Raspberry PI usado por el equipo de Smart Parking, esta etapa es innecesaria puesto que ya esta habilitada como servidor.

En caso de disponer de una Raspberry PI reseteada, os recomiendo el siguiente enlace para poder usarla como servidor web.

<https://pedropablomoral.com/raspberrypi/proyectos/servidor-casero/>

También existe una presentación del CEI de un seminario de Raspberry PI que os puede ser de ayuda. Preguntad a vuestro tutor a ver si os puede dar acceso a la presentación “Primeros pasos en Raspberry PI”.

2. Entorno de desarrollo

La página web esta programada en HTML con inyecciones de PHP para la parte dinámica. También cuenta con un fichero CSS para mejorar la estética de la aplicación.

En nuestro caso la aplicación ha sido desarrollada directamente desde la Raspberry PI. Al tratarse de una web sencilla esto no supuso demasiados problemas. Sin embargo, si se quiere hacer aplicación completa y funcional, creo que seria recomendable realizar el desarrollo desde un ordenador estándar con mucha más potencia y ergonomía que una Raspberry PI y al final migrar el proyecto a la Raspberry.

Para la base de datos se ha utilizado phpMyAdmin que es una herramienta que permite tener una base de datos local en MySQL. Las claves de acceso a nuestra cuenta son:

- user1
- password1

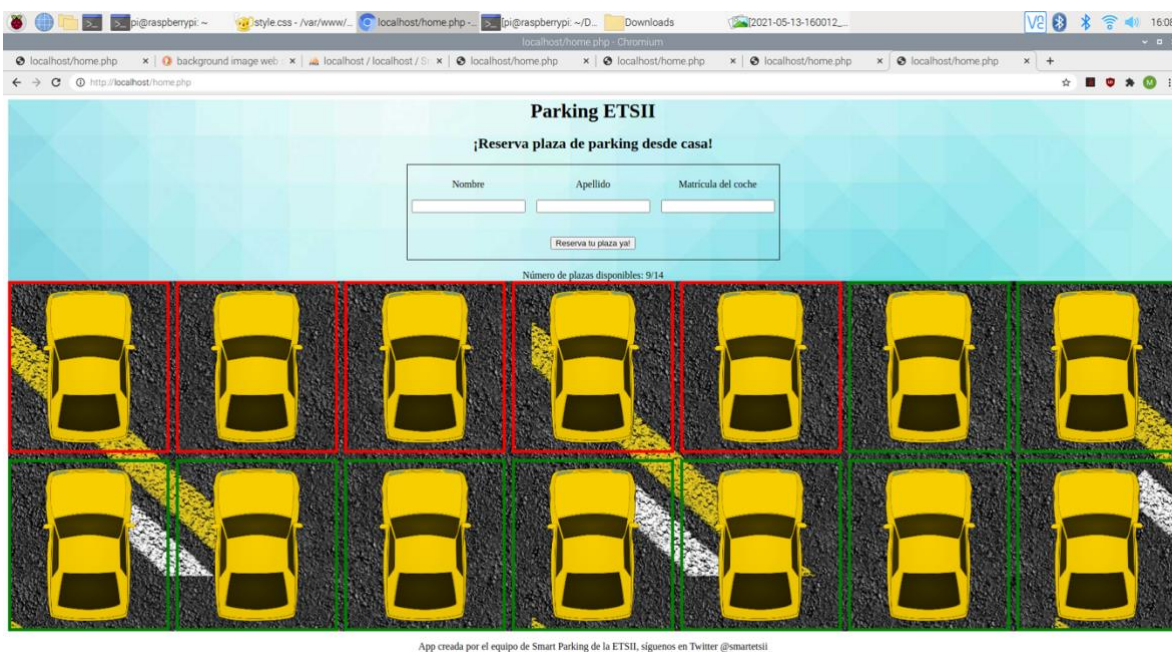
En nuestro caso hemos usado una única tabla llamada “Reservas” donde cada vez que un usuario realizaba una reserva se añadía una fila en dicha tabla con la información del formulario: nombre, apellido y número de matrícula de coche.

En nuestro caso se intento implementar algún sistema para acceder al historial de errores de la página web y de este modo saber porque y donde estaba fallando el código cuando saltaba un error. Sin embargo, esto no se consiguió por lo que la página se programo “a ciegas”. De nuevo, para una aplicación sencilla esto no supone demasiado problema, pero si se quiere hacer algo serio, considera de gran utilidad tener acceso al historial de errores.

3. Funcionamiento de la página web

Una vez habilitada la Raspberry PI como servidor, solo queda descargar los ficheros que componen la página web (<https://github.com/marco-novaes98/smartParking>) y moverlos a la carpeta html de la Raspberry PI (ruta: /var/www/html). Ya podemos lanzar nuestra aplicación, basta con abrir el navegador y buscar <http://localhost/home.php>.

A. Home



El fichero home.php es el corazón de la aplicación. La mitad superior de la página contiene el formulario (nombre, apellido y matrícula de coche) y el botón para reservar una plaza de parking.

En la mitad inferior encontramos un contador del número de plazas disponibles y un tablero dinámico compuesta por 2 filas y 7 columnas (14 casillas). Cada casilla corresponde a una plaza del parking y el color del borde de la casilla informa de la disponibilidad de la plaza (verde para disponible y rojo para ocupada). Supongamos que el parking esta vacío (todas las casillas en verde). La llegada sucesiva de coches hará que las

casillas se vayan progresivamente cambiando al color rojo, este cambio sucederá en el sentido de lectura (de izquierda a derecha, de arriba a abajo). Este orden es independiente de la ubicación real de las plazas ocupadas en el parking, lo único importante es que el número total de plazas disponibles sea el mismo en el parking como en la aplicación.

Cada vez que se abre la aplicación, se cuenta el número de líneas de la tabla Reservas. Restando al número total de plazas (14) este número de líneas, obtenemos el número de plazas disponibles. Este número se utiliza para actualizar el contador de plazas y para colorear en verde el número de plazas correspondiente.

Al rellenar el formulario y clicar en el botón “Reserva tu plaza ya!” ocurren dos cosas:

- Se añade en la base de datos (phpMyAdmin) una nueva fila con los datos del formulario
- Se redirecciona al usuario a la página de confirmación de la reserva

B. Confirmación



El fichero confirmacion.php es muy básico. Recibe el nombre y la matrícula del usuario de la reserva vía la url (ej: <http://localhost/confirmación/?nombre=Marco&matricula=123>) y dispone un mensaje personalizado en el que se indica que la reserva se ha realizado correctamente y se recuerdan las “reglas” de la reserva. Las reglas expuesto por ahora son arbitrarias, se han puesto simplemente para dar un aspecto más realista a la aplicación. En la versión final habría que reflexionar sobre la información pertinente para los usuarios antes y después de realizar la reserva.

C. Style

El fichero style.css es simplemente unas pocas reglas CSS para mejorar la estética de la aplicación. Este fichero se aplica tan solo a home.php, confirmación.php tiene la parte estética implementada directamente en HTML. Esto último no se suele recomendar por lo que sería pertinente crear otro fichero CSS para el archivo confirmación.php.

4. Tareas por hacer

La versión actual de la aplicación es muy limitada, para ofrecer un servicio completo y de calidad quedan numerosas cosas por hacer. A continuación, voy a dar una lista de algunas ideas que en mi opinión merecería la pena implementar en la aplicación:

- Mejora de la estética
- Crear una tabla en phpMyAdmin con todas las reservas (datos del usuario, estado de la reserva, número de identificación único...)
- Informar al usuario de las condiciones relativas a las reservas
- Añadir más campos al formulario (mail, numero de teléfono...)
- Información de contacto con el equipo (mail, acceso directo a redes sociales...)
- Mail de confirmación con número de referencia al realizar una reserva
- Posibilidad de registrarse e iniciar sesión como usuario para tener acceso a tu información personal (historial de reservas, reservas en curso...)
- Posibilidad de cancelar una reserva (por ejemplo desde el espacio usuario)
- Estudiar la posibilidad de poder acceder a la web desde alguna otra de las plataformas de la ETSII
- Si tras el tiempo concretado en las condiciones de la reserva el coche de la reserva no aparece, habrá que liberar la plaza reservada y enviar un correo para informar al usuario que su reserva a caducado por exceso de tiempo

A parte de las mejorar de la web hay otros aspectos vitales para que el proyecto pueda ofrecer un servicio operativo.

Por ahora la aplicación esta en local y solo se puede acceder desde la Raspberry PI. Hay que ofrecer un acceso remoto para que todo el mundo la pueda usar. Para ello supongo que habrá que comprar un dominio web y usar algún tipo de herramienta para migrar la aplicación local a ese dominio web. No doy más detalle ya que desconozco este proceso.

Por último, queda el acoplamiento de la página web con nuestro sistema inteligente de detección de coches. Tras la llegada de un coche, el dispositivo inteligente deberá comunicar la matricula a un algoritmo que se encargue de ejecutar el procedimiento requerido en función de la matrícula del coche:

- Si la matrícula corresponde a una reserva vigente, se tendrá que actualizar el estado de la reserva y notificar al usuario que el sistema ha detectado correctamente su llegada al parking
- Si la matrícula no corresponde a una reserva y quedan plazas disponibles, habrá que actualizar la base de datos para indicar que hay un coche nuevo en el parking y que el usuario no ha hecho uso de la aplicación de reservas
- Si la matrícula no corresponde a una reserva y todas las plazas libres están en realidad reservadas, habrá que comunicar de algún modo al usuario que no tiene permitido aparcar en esa plaza de parking

5. Conclusión

Espero que esta breve documentación pueda ser de ayuda y no dudéis en contactarme (marco.novaesm@alumnos.upm.es o marco.novaes.mz@gmail.com) si tenéis cualquier duda. Si finalmente un equipo retoma este proyecto y consigue llevar la aplicación a buen puerto, me encantaría que me mandaseis el link a mi correo para poder ver vuestro trabajo ¡Suerte con el proyecto!