



Tabla de contenido

[1 Descripción de la aplicación 3](#_Toc533533595)

[2 Diseño de la interfaz y lógica de negocio 4](#_Toc533533596)

[2.1 Pantalla Login 4](#_Toc533533597)

[2.2 Pantalla Inicio 5](#_Toc533533598)

[2.2.1 Inicio para profesores 5](#_Toc533533599)

[2.2.2 Inicio para usuarios 6](#_Toc533533600)

[2.2.3 Navegación y menú 7](#_Toc533533601)

[2.3 Pantalla Ranking 8](#_Toc533533602)

[2.4 Pantalla Escanear 9](#_Toc533533603)

[2.5 Pantalla Premios 10](#_Toc533533604)

[3 Servicios e Información Persistente 11](#_Toc533533605)

[3.1 Información de la base de datos 11](#_Toc533533606)

[3.2 Información local 14](#_Toc533533607)

[3.3 Servicios 14](#_Toc533533608)

# Descripción de la aplicación

La aplicación busca la motivación del uso de la bicicleta como medio de transporte limpio dentro de nuestro alumnado. Para ello, se diseña buscando el concepto de gamificación.

A cada alumno se le entrega un código QR (<http://www.codigos-qr.com/generador-de-codigos-qr/)> en el cual se han rellenado los campos de nombre y cargo (curso). Este código qr lo llevarán los alumnos en su bicicleta a modo de matrícula. Cada viaje que hagan con la bicicleta le supondrán puntos que podrán canjear por premios.

Existen dos roles:

Profesor: que puede editar a los alumnos, escanear los código qr para otorgar puntos y aprobar los premios que los alumnos soliciten.

Alumno: que podrá ver el ranking y solicitar canjeo de puntos por premios.

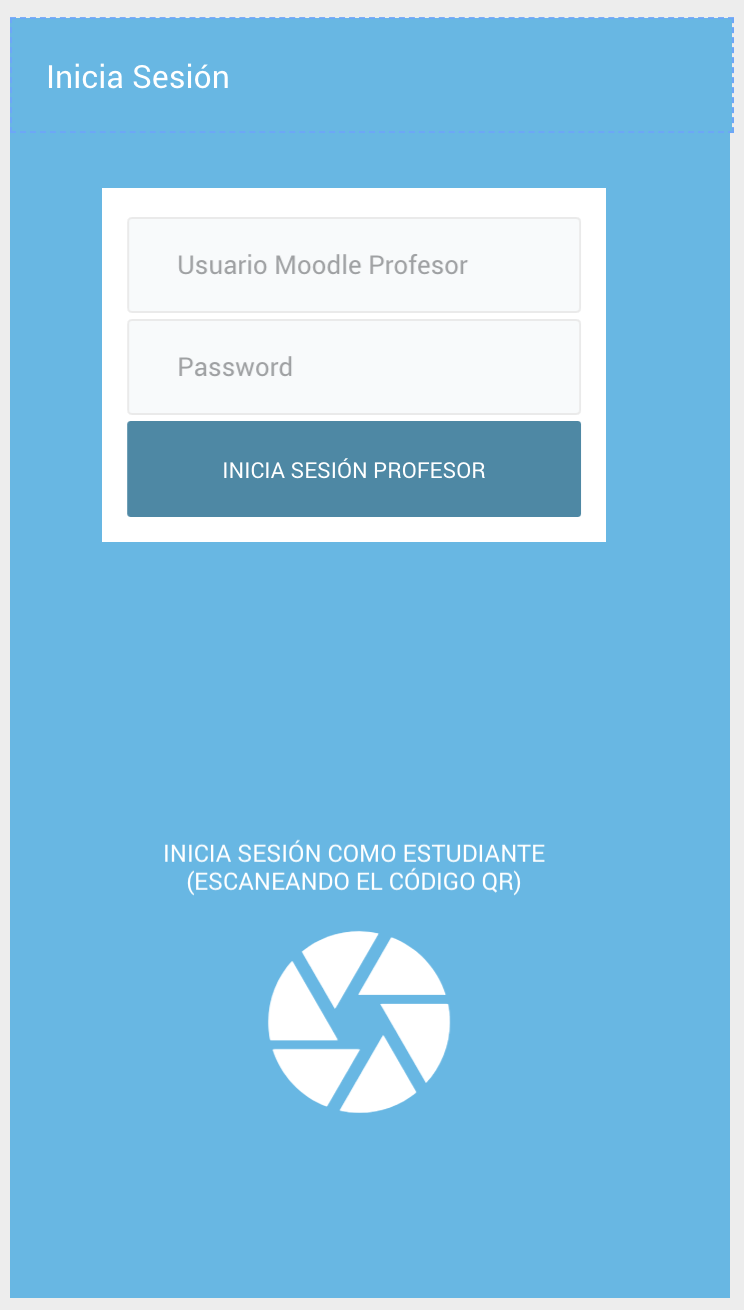
# Diseño de la interfaz y lógica de negocio

## Pantalla Login

Al inicio se realiza una comprobación de sesión creada en base de datos local. Si existe se cargan los datos y se pasa a la pantalla inicio (correspondiente al rol adecuado). Si no existe muestra la pantalla de login.

La pantalla de login permite dos tipos de procesos:

* Login como profesor que consiste en un formulario en el que se emplean las credenciales de la Moodle del centro. Se realiza una petición http contra un servicio REST y se guarda el token en local para recordar el inicio junto con los parámetros por defecto.
* Login como estudiante que consiste en un escáner de código QR que extrae la información (nombre y curso) y realiza una consulta en la base de datos (CloudStorage). Si el usuario existe guarda en local el token para recordar sesión junto con los parámetros personalizados del usuario. Si no existe, lo crea (con los puntos y premios adecuados, junto con los parámetros por defecto de usuarios) y guarda la información necesaria para recordar sesión.



## Pantalla Inicio

Existen dos roles (profesor y usuario). En función del rol se muestra una pantalla de inicio u otra.

### Inicio para profesores

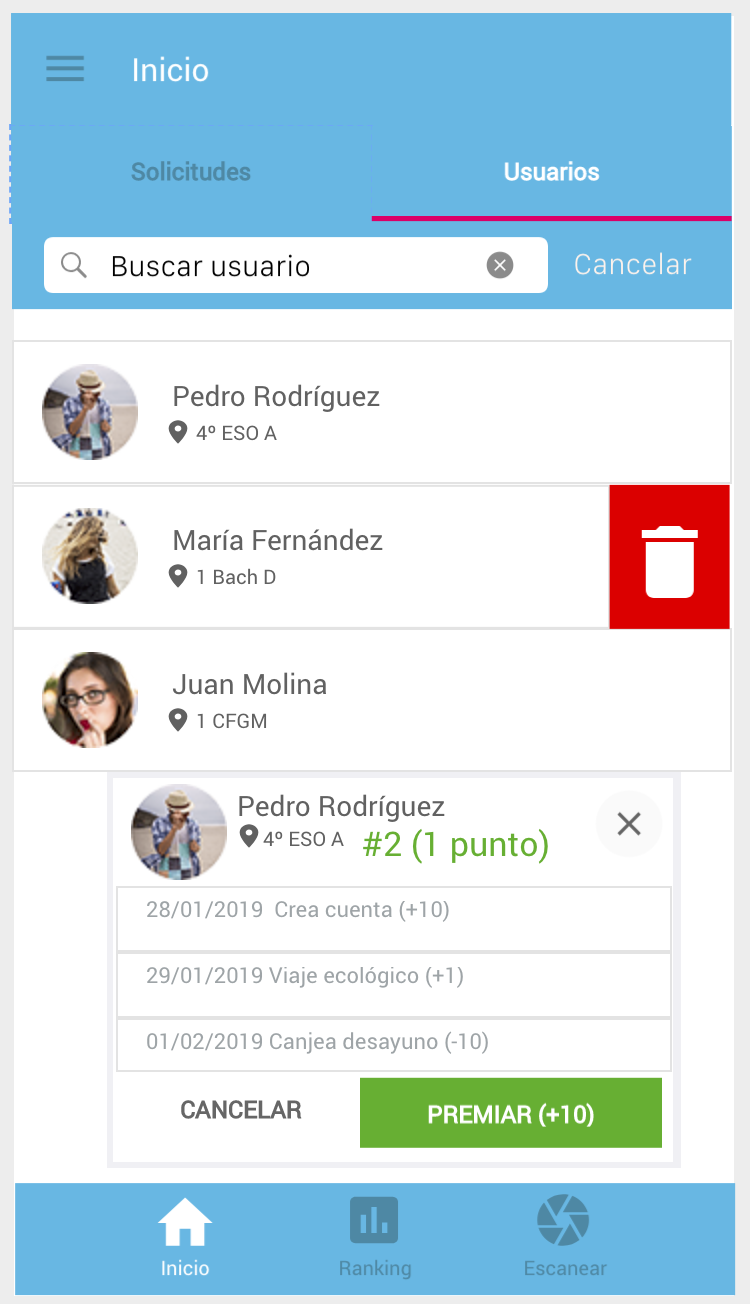
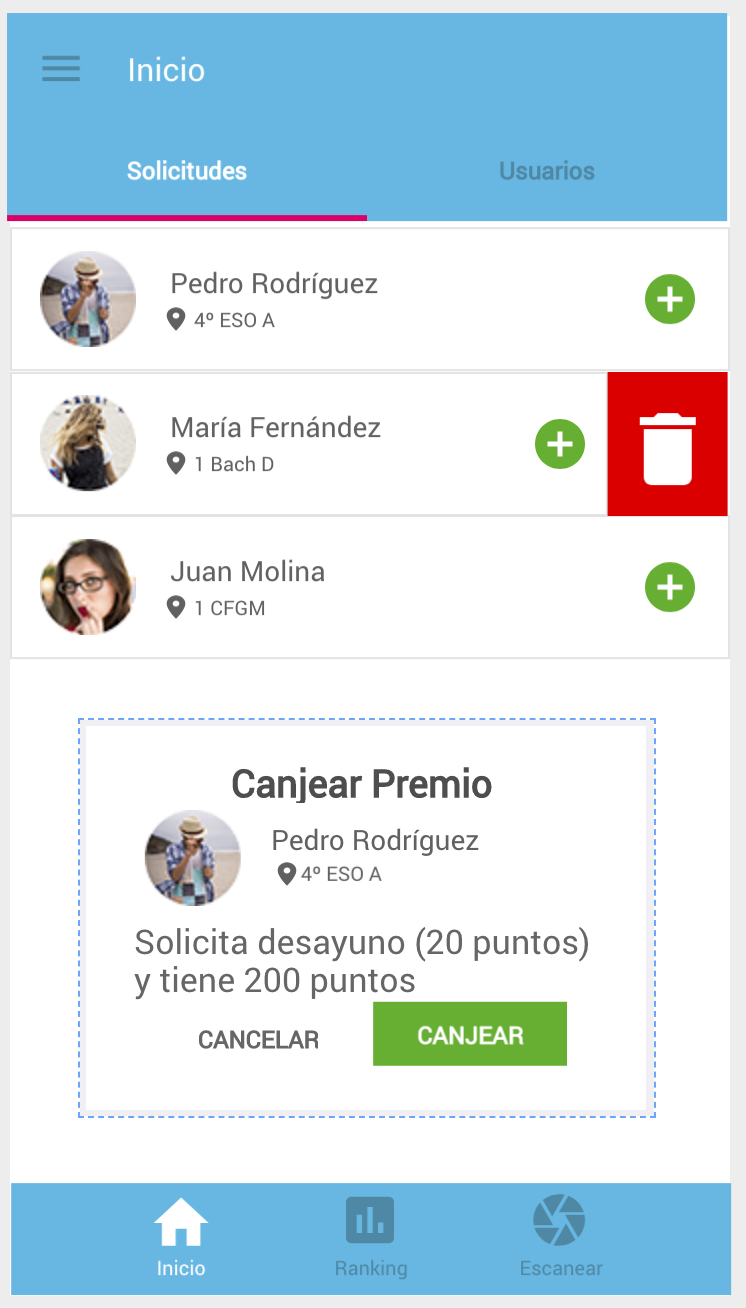
Existe dos segmentos: Solicitudes (seleccionado por defecto) y Usuarios.

* En el segmento de solicitudes se muestran todas las peticiones de canjeo de puntos realizadas por los usuarios. Se muestra en forma de lista deslizante (si se desliza se muestra la cancelación de la petición con confirmación) con el avatar y la información básica del usuario.

En caso de pulsar sobre la petición se abre un modal con la información relativa a la petición (puntos) y la posibilidad de realizar el canjeo efectivo que actualizará los puntos finales y notificará al usuario su aprobación.

* En el segmento de usuarios aparecen todos los usuarios del sistema con la posibilidad de realizar búsquedas. Es una lista deslizante con la opción de borrado de usuario por confirmación.

Si se pulsa un usuario se abre un modal con toda la información del usuario y su historial junto con la posibilidad de premiar.

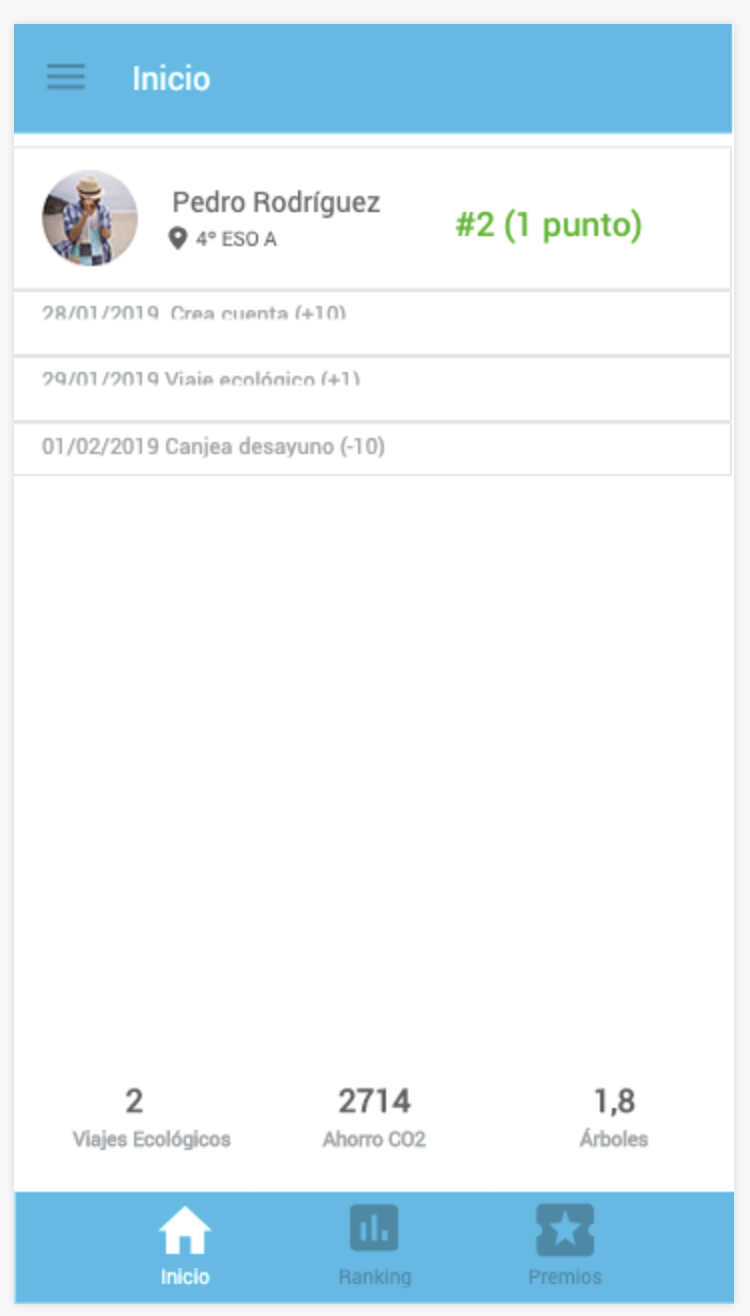


La navegación entre segmentos se realiza por deslizamiento o por click en el botón de título de segmento.

### Inicio para usuarios

Se muestra información de identificación del usuario con un resumen de sus puntos y ranking.

Se realiza un listado del historial y como footer un resumen del impacto ecológico de su actividad.



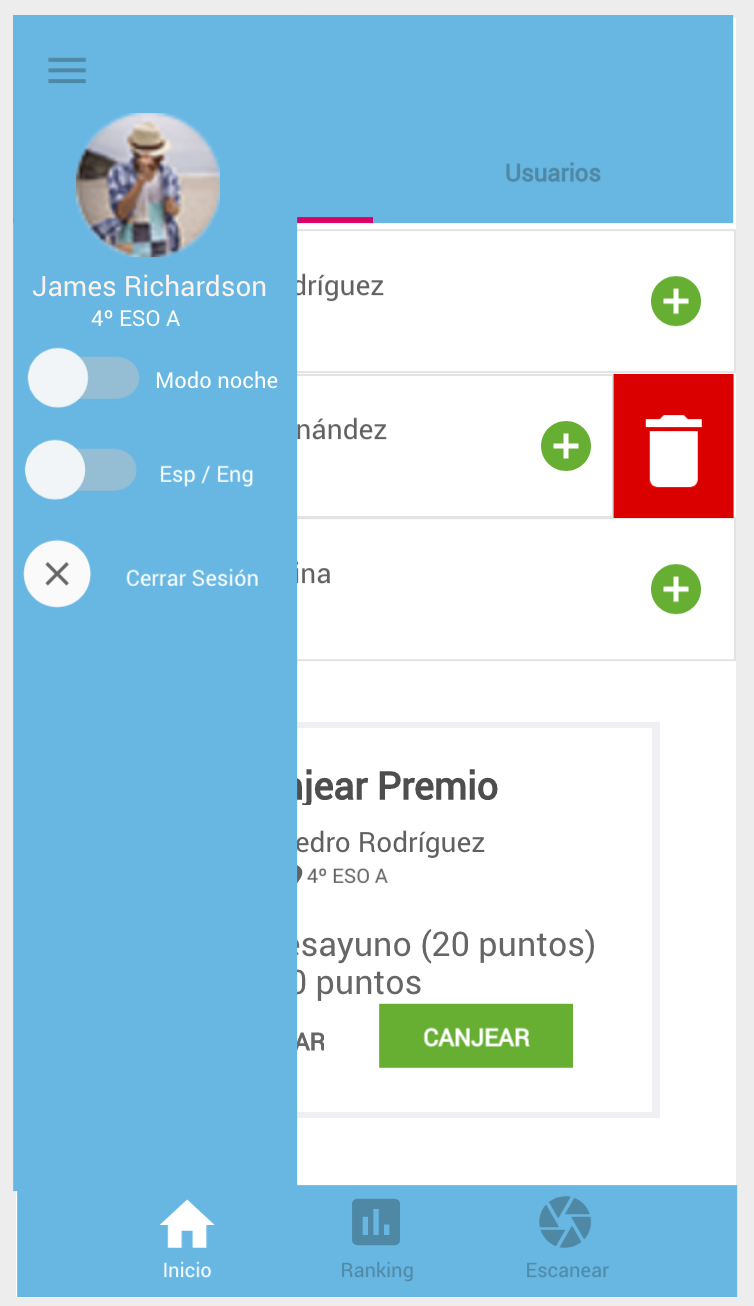
### Navegación y menú

Tanto para profesores como para alumnos la navegación se realiza por tabs y existe un menú desplegable desde la izquierda. En dicho menú se muestra el avatar que pulsando sobre él se podrá editar seleccionando una foto de la galería o de la cámara.

Muestra la información de identificación de usuario y la posibilidad de activar la interfaz en modo nocturno y con idioma inglés.

Todas estas opciones se guardan en local y se actualizan en la base de datos del usuario para recordarlas en caso de cierre de sesión.

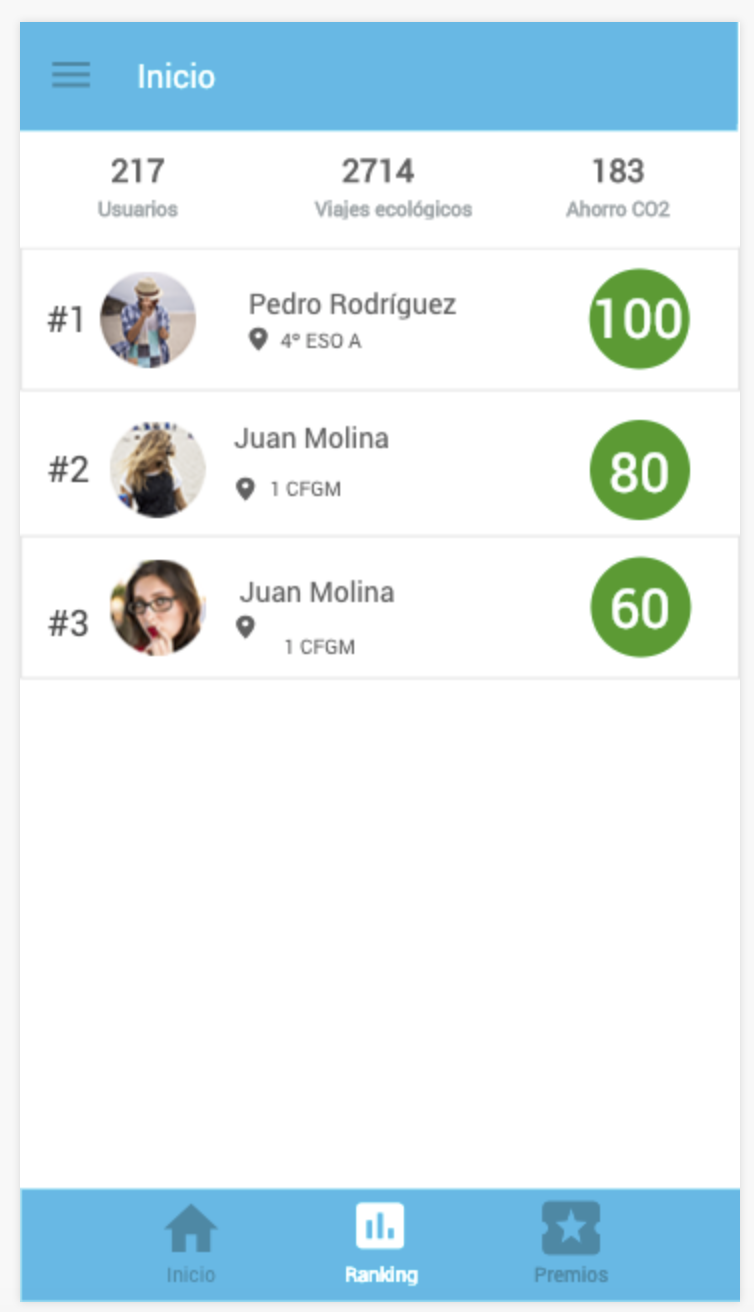
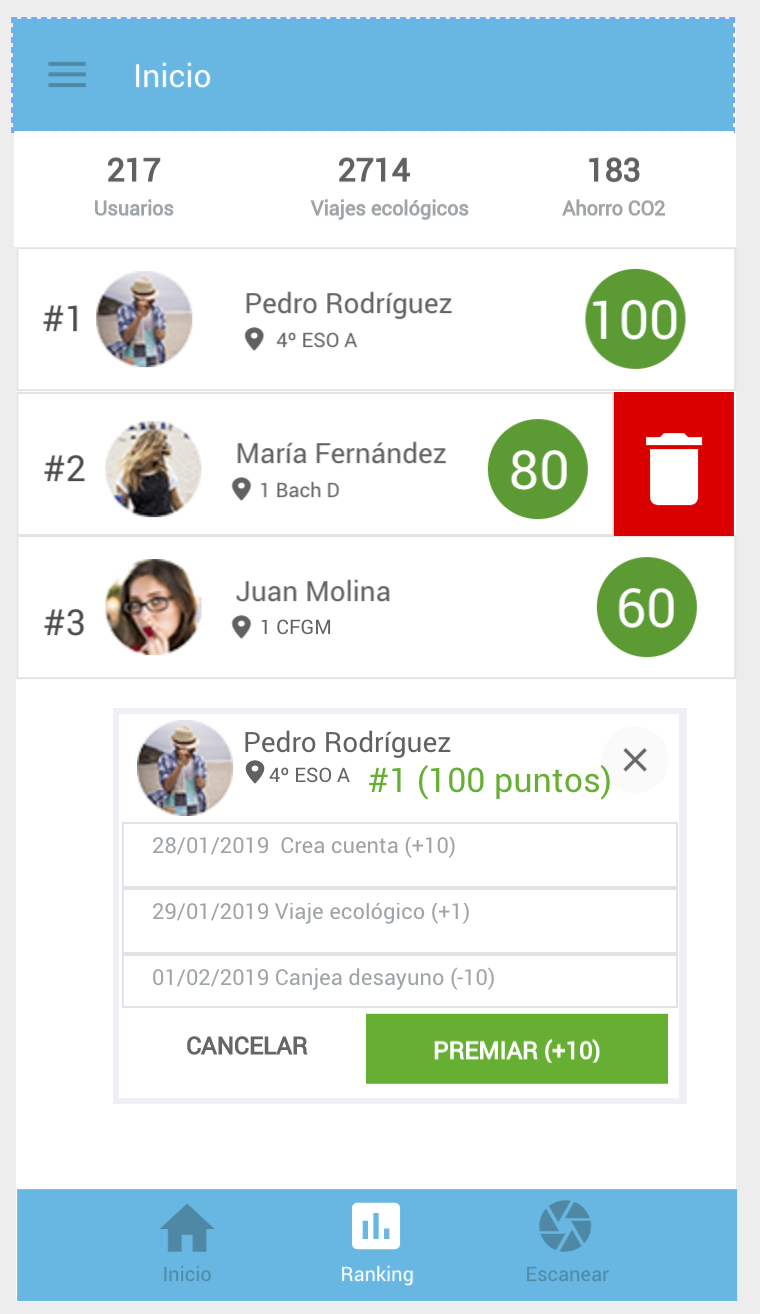
Finalmente da la posibilidad de cerrar sesión.



## Pantalla Ranking

Para el rol de profesor (imagen izquierda) en la pantalla Ranking se muestra una lista deslizante ordenada por los puntos conseguidos. En dicha lista se muestra la información de los usuarios y la posibilidad por deslizamiento del borrado con confirmación. En caso de pulsar un usuario aparece el modal de usuario con su historial y la posibilidad de premiar. En la cabecera aparece un resumen del impacto ecológico del sistema completo.

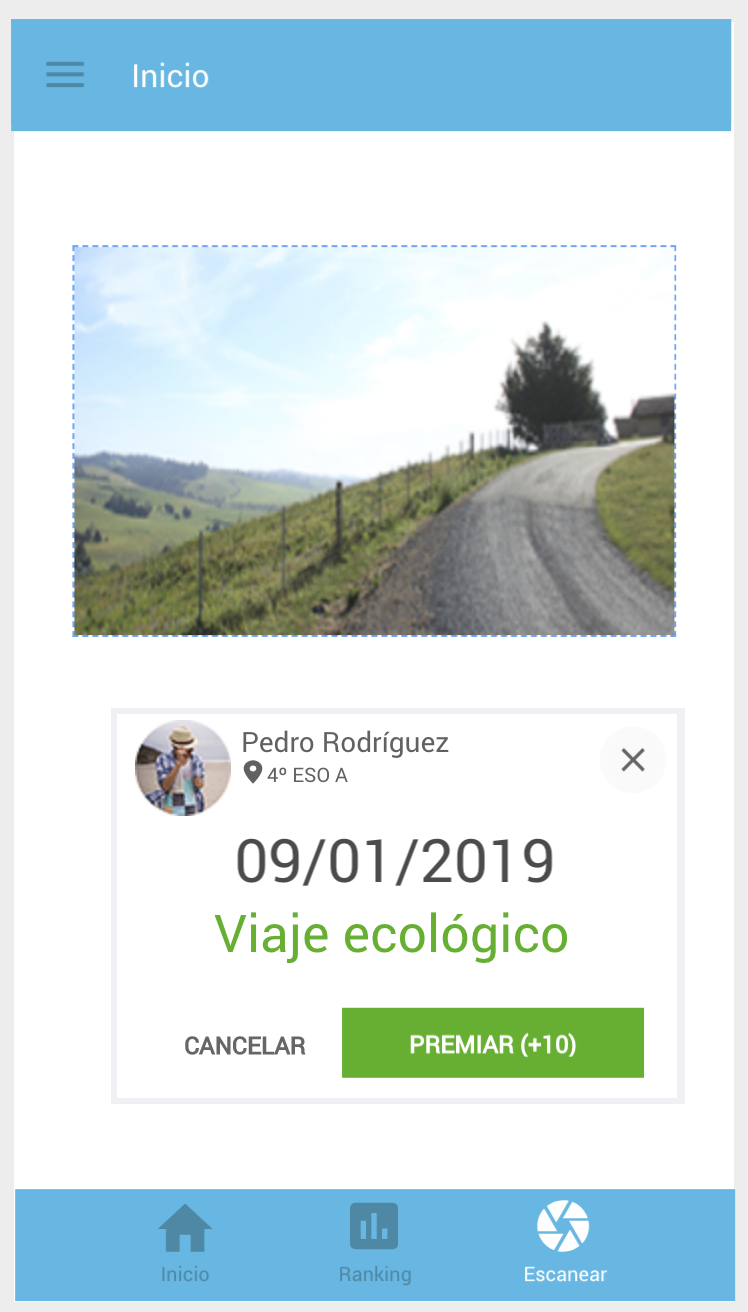
En el caso de rol de usuario (imagen derecha) la ventana es igual, pero sin posibilidad de borrado ni edición por pulsado sobre usuarios.



## Pantalla Escanear

Para el rol de profesor la tercera pestaña es la posibilidad de escanear los códigos QR de las bicicletas del aparcamiento.

Aparece la cámara, cuando se realiza la lectura del código aparece un modal con la información de identificación del usuario y se solicita confirmación de actualización de puntos. En caso de que exista una entrada de puntos el mismo día se realiza un aviso por confirmación para no escanear dos veces un mismo QR el mismo día por equivocación.

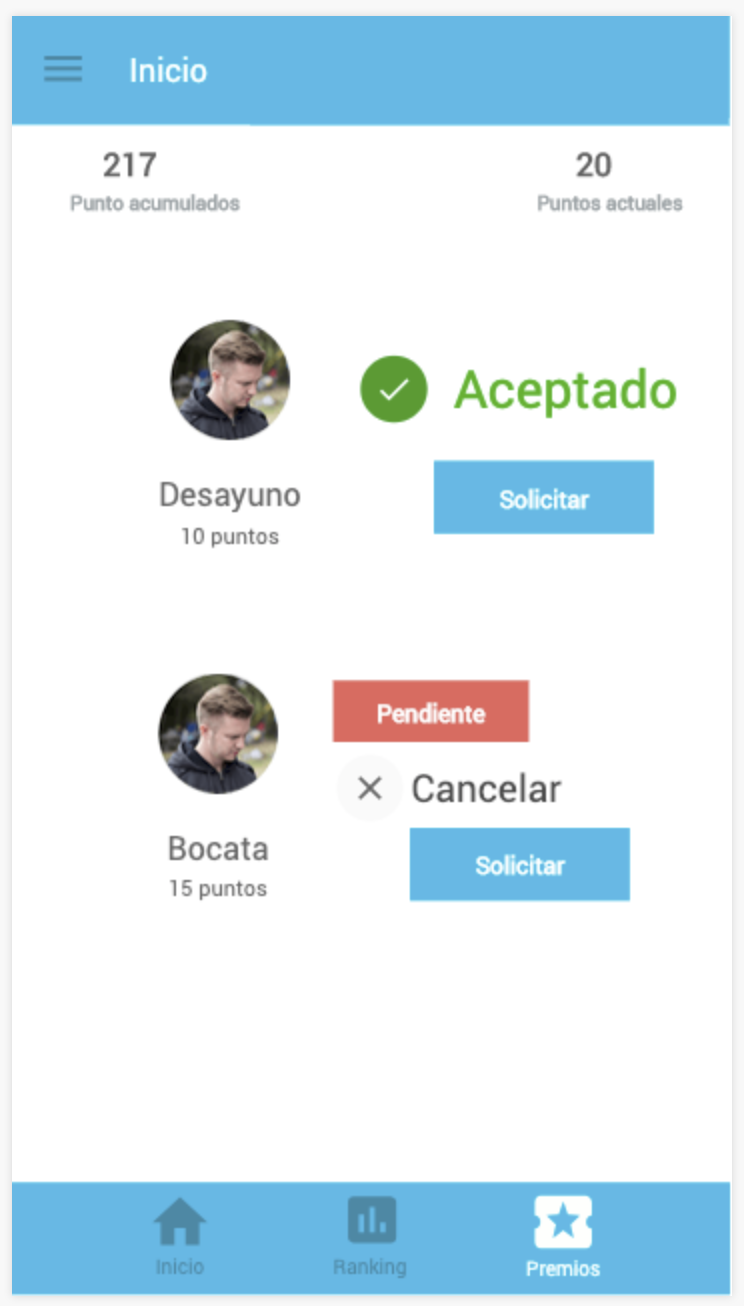


## Pantalla Premios

En el rol de usuario la tercera pestaña es la de premios. En primer lugar aparece un resumen sobre los puntos del usuario. Luego un menú por cards sobre los premios a solicitar (solo activos aquellos que se alcancen con los puntos actuales).

En caso de solicitar un premio se coloca como pendiente y se realiza una reserva de puntos temporal. En este estado (antes de que se confirme la aceptación del premio) se podrá cancelar una petición.

Si el profesor acepta el premio aparecerá como aprobado durante los próximos 7 días.

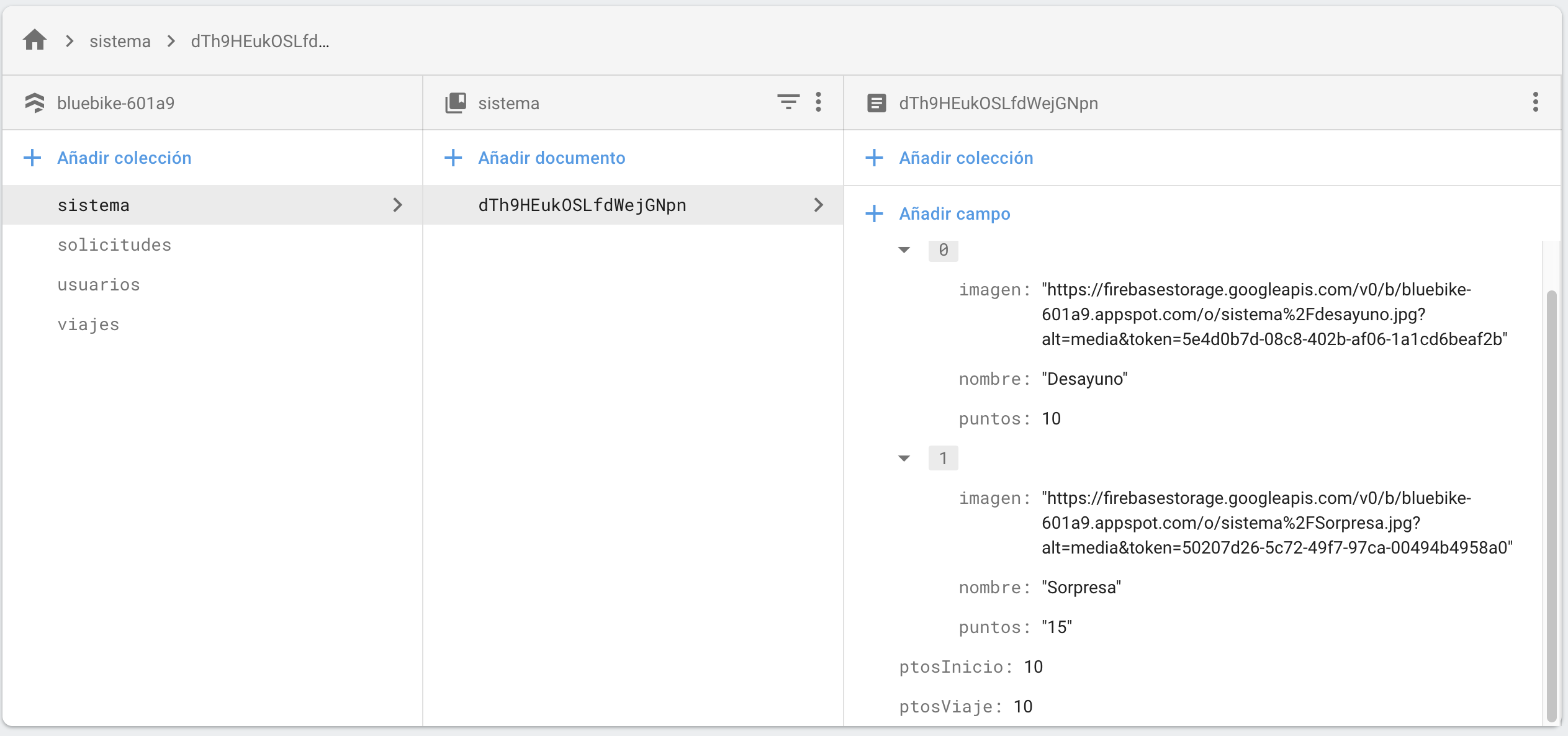


# Servicios e Información Persistente

## Información de la base de datos

El sistema de almacenamiento que se emplea es Cloud Storage de Google. Este sistema de base de datos NoSQL se organiza por colecciones. Para esta aplicación emplearemos cuatro colecciones:

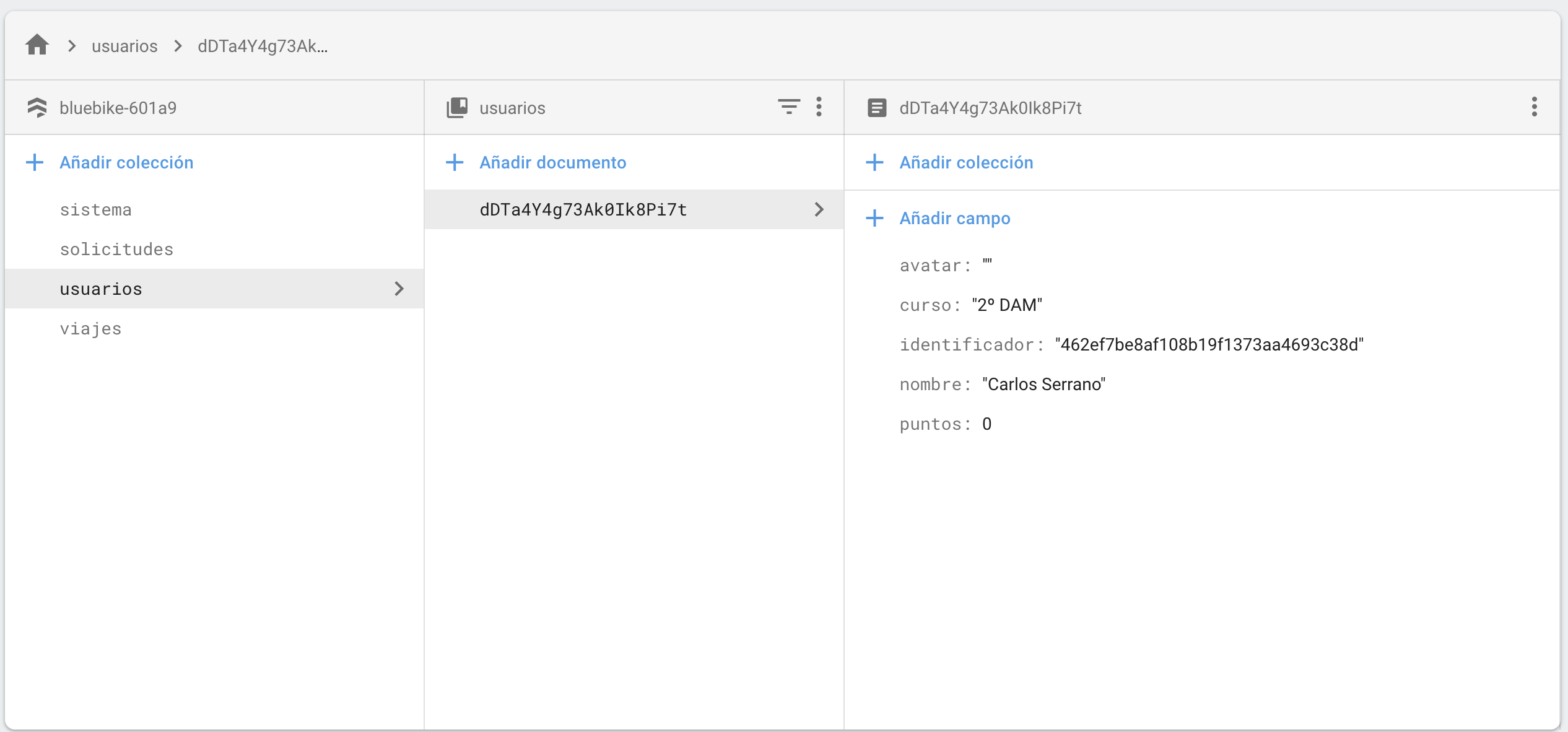
**sistema**



Tendrá una entrada única que será la configuración básica del sistema. Aparecen campos como:

* ptosInicio: son los puntos que se regalan cuando un nuevo usuario se registra en el sistema.
* ptosViaje: son los puntos que se regalan por cada viaje.
* premios: es un array con los diferentes premios que se pueden solicitar, tienen tres campos: imagen: url del storage para el icono, nombre descriptivo y puntos que vale.

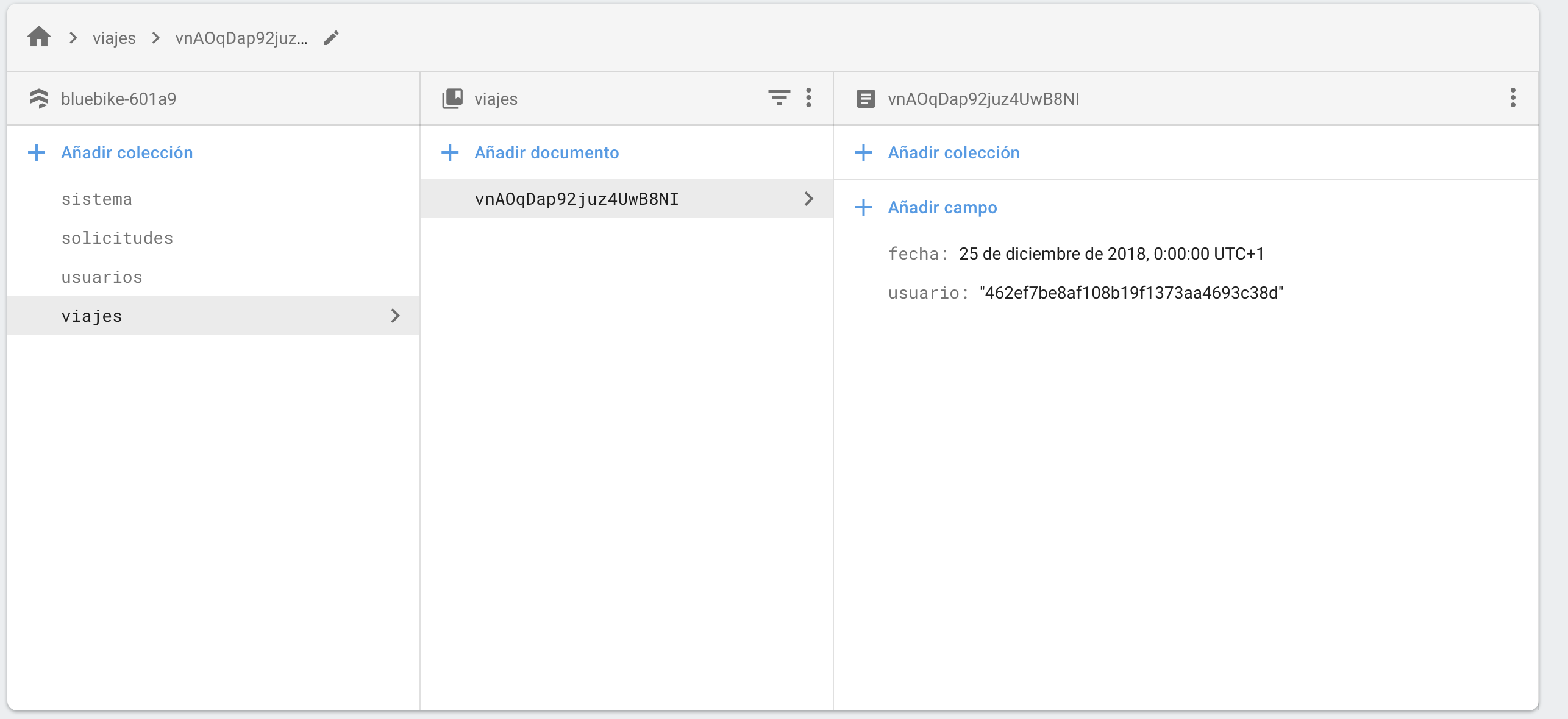
**usuarios:**



Se emplea para almacenar los datos de los usuarios registrado:

* avatar: es la url de la imagen de perfil que se almacena en Storage.
* curso: extraído del código QR.
* nombre: estraído del código QR.
* identificador: es la relación unívoca entre un código QR y una entrada de esta base de datos. Se calcula como le MD5(“nombre(curso)”).
* puntos: son los puntos actuales disponibles.

**viajes:**



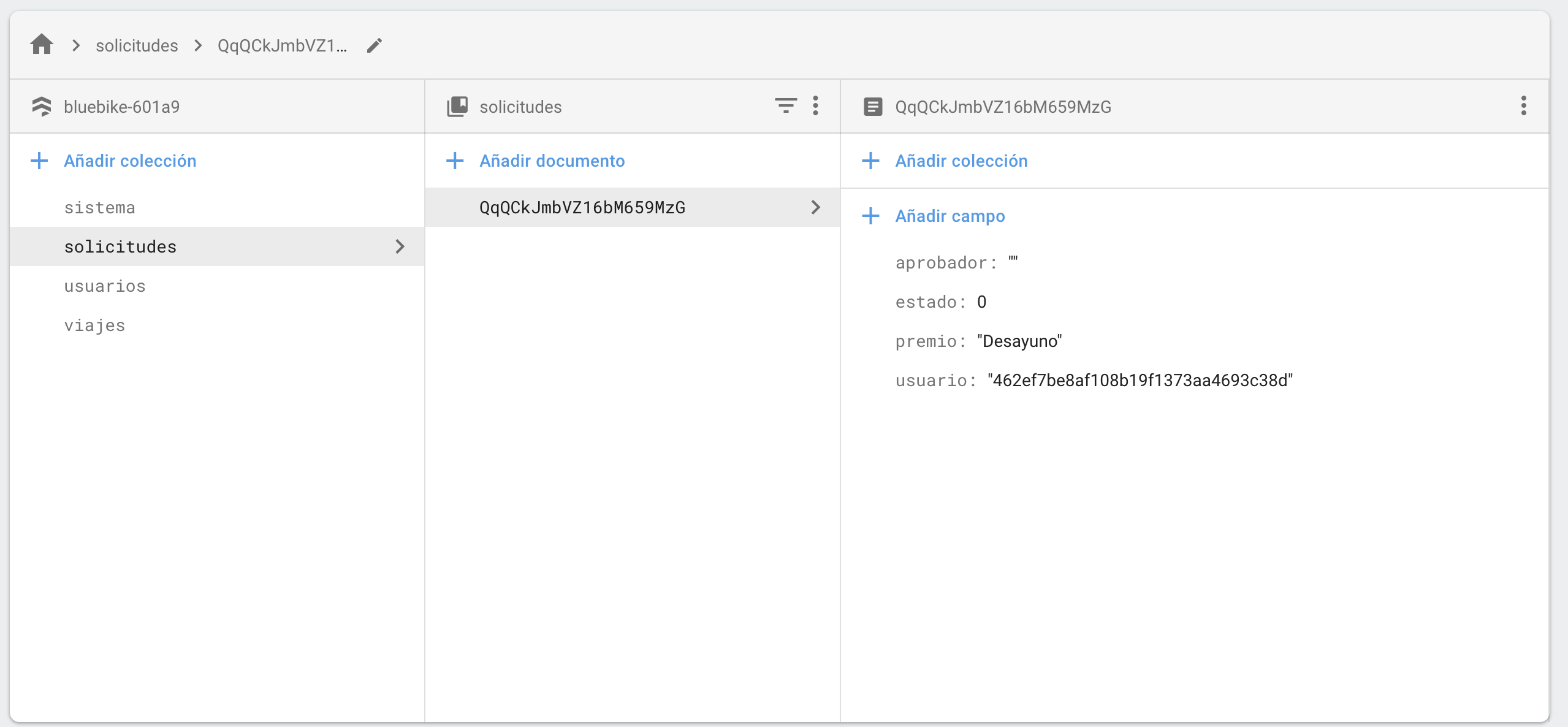
Consta de dos campos:

fecha: fecha y hora del escáner que justifica el viaje.

usuario: identificador único del usuario (extraído del código QR como MD5(“nombre(curso)”) ).

Esta tabla almacena todo el historial, con lo cual sirve para calcular todos los puntos conseguidos por un usuario, simplemente multiplicando el número de viajes por el número de puntos que vale un viaje.

**solicitudes:**



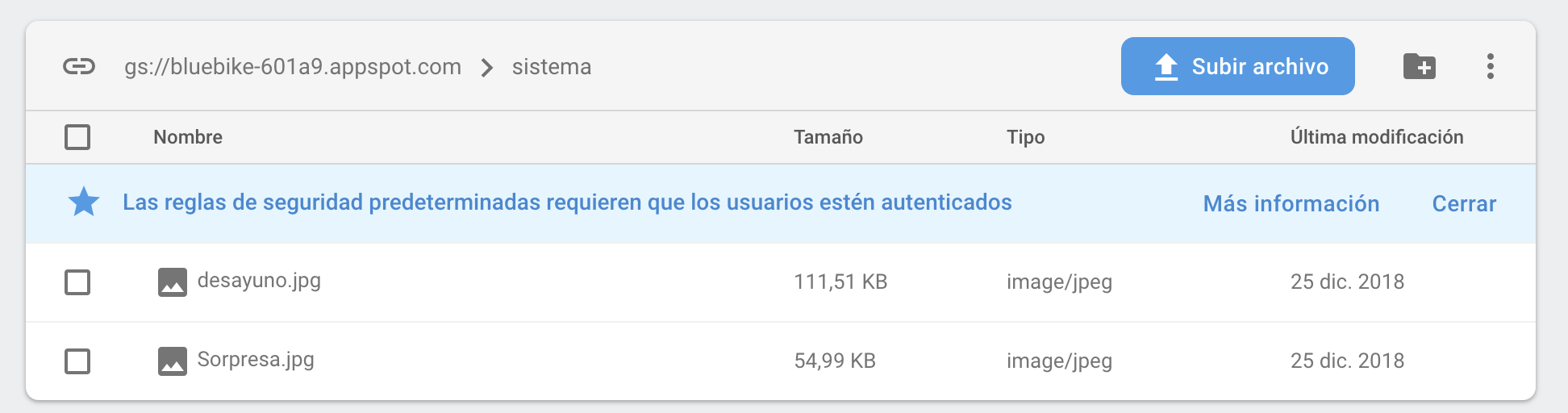
Esta tabla almacena el historial de solicitudes de premios y su estado. Consta de 4 campos:

* usuario: identificador único del usuario que lo solicita.
* premio: nombre del premio que concuerda con las variables del sistema para calcular cuántos puntos vale.
* estado: indica el estado de la solicitud (0= solicitada y pendiente de aprobación, 1=aprobada)
* aprobador: email del profesor que ha aprobado la solicitud.

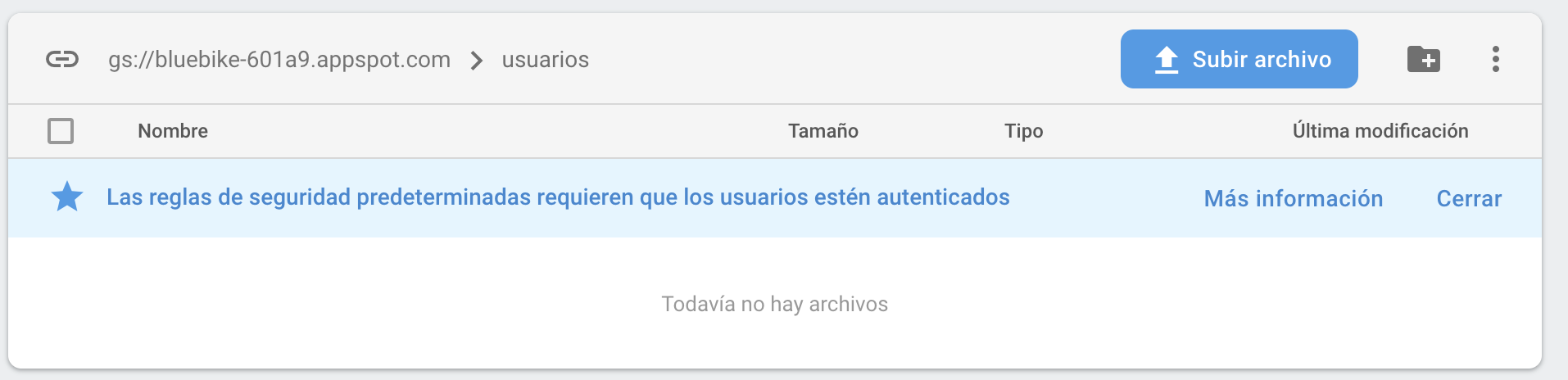
Un usuario solo puede tener una solicitud en pendiente de aprobar.

En cuanto a Storage, existen dos carpetas:

sistema: que almacena imágenes del propios sistema, como son los iconos de los premios.



usuarios: que almacena los avatares de los usuarios.



## Información local

En cuanto a la información almacenada en el propio dispositivo, se emplea sqlite de la librería Storage de Ionic. Entre los campos más relevantes, podemos destacar:

* logged= true | false (false por defecto): identifica si un usuario está loggeado o no, en caso de que no esté “logueado” se cargan los datos por defecto del sistema.
* identificador= identificador único en caso de usuario | mail en caso de profesor (null por defecto): se emplea para identificar al usuario o al profesor.
* role: prof | user (user por defecto)
* idioma= es | en (es por defecto): la interfaz es multilingüe.
* skin= light | dark (light por defecto): la interfaz es configurable en dos esquemas de colores.
* tutorial = on | off (on por defecto): por defecto un usuario que no acaba de loguearse se le muestra un tutorial (“skippable”). En caso de que el usuario indique que no quiere volver a ver el tutorial no se mostrará nunca más.

Todas estas variables son locales y no se cargan en ningún momento en base de datos.

## Servicios

Para autenticar a los usuarios emplearemos un scanner de código QR que nos permitirá calcular su identificador único (MD5(“nombre(curso)”)).

Para autenticar s los profesores emplearemos un servicio REST disponible por la Moodle de nuestro centro. La url del endpoint es:

[https://moodle.iesfranciscodelosrios.es/login/token.php?username=[USUARIO]&password=[Password]&service=PARTES](https://moodle.iesfranciscodelosrios.es/login/token.php?username=%5bUSUARIO%5d&password=%5bPassword%5d&service=PARTES)

Antes de realizar la autenticación pondremos como restricción que el mail del profesor debe empezar por ‘p’. El servicio devuelve:

* En caso correcto:

{"token":"3c78e763bc264bb7e4e8074157a8ae9c","privatetoken":"mesMckyST7GqCXlzpIyvzHnKVSJO1JmaE6Cqf6PZFjf0LhOckkkgxdBWggFEaz18"}

* En caso erróneo:

{"error":"Datos err\u00f3neos. Por favor, int\u00e9ntelo otra vez.","errorcode":"invalidlogin","stacktrace":null,"debuginfo":null,"reproductionlink":null}