

## INFORME SEGUNDO PARCIAL

VELOBIKE es una empresa que se dedica a las bicicletas. Su función principal en la app es registrar y mostrar los recorridos realizados en bicicleta dentro de la misma. Esta aplicación fue desarrollada con Vue cli y Vuetify. Se ha implementado Vue para desarrollar las principales pantallas de la app.

**App.vue** (root) contiene un menú navigation-drawer que permite mostrar los títulos y sus paths, junto con el footer también incorporado desde Vuetify. Incorpora las vistas a través de la etiqueta **router-view** dentro de un transition que hace un efecto gracias a la dependencia **Animate.css**

A través de **vue-router** se crearon 3 vistas:



La **HOME** fue incorporada desde las vistas, además de vista es componente. La única funcionalidad de HOME.VUE es hacer el **FETCH** a la API de Velobike y así también la guarda en Local Storage.

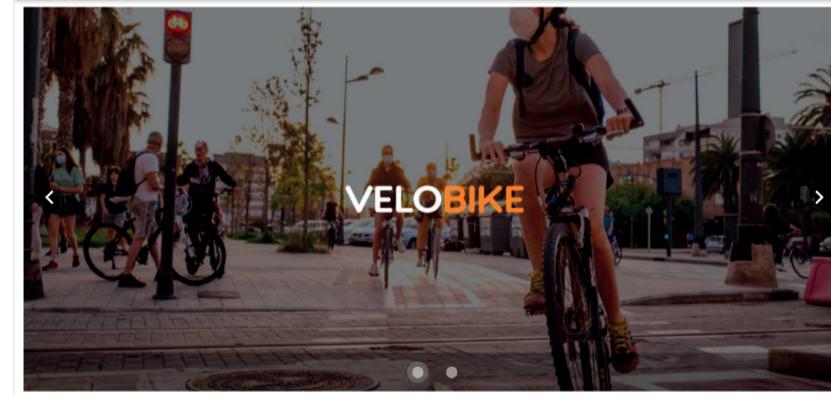
La API Velobike contiene 3 recorridos a modo de ejemplo mostrando lo más “común” a la hora de registrar uno de los recorridos.

En recorrido y muestra de la API se aplican clases por medio del **v-bind** con un operador ternario.

En caso de que sea un recorrido de Montaña, muestra una imagen y si es Ciudad, muestra otra.

En esa vista se incorporaron dos **componentes visuales de vuetify**:

- El Carousel a modo de inicio
- Las cards de “beneficios” y en la muestra de nuestra API.



### Bienvenido a VELOBIKE

Ingresá todos tus recorridos en bicicleta y llevá un registro de cada uno. En Velobike queremos ayudarte a llevar una vida saludable.  
Ingresando todos los detalles del recorrido podés llevar un registro ordenado y dividido entre montaña y ciudad.

+ EMPEZAR CON LOS RECORRIDOS

### Beneficios de andar en bicicleta



#### Posición de la espalda

La inclinación de nuestro cuerpo hacia el manubrio favorece que los músculos del área lumbar se tensionen y se refuerzen, de esta forma



#### Fortalece músculos

Fortalece las rodillas, músculos y protege articulaciones. Además, al recorrer la mayor parte del peso del cuerpo en el asiento, las rodillas no



#### Reduce el estrés

El ejercicio físico permite oxigenar el cerebro y crear determinadas hormonas que ayudan a combatir estos estados negativos. Montar en

### Algunos de los recorridos más conocidos



**“INGRESAR.VUE”** también es una vista y componente con la funcionalidad de ingresar los recorridos de los usuarios.

El formulario se valida según los datos ingresados a la hora de mandar el formulario.  
Todos los campos son obligatorios.

The screenshot shows a form titled "Ingresar recorridos". It has four input fields: "Nombre" (Name), "Distancia en km" (Distance in km), "Duración en minutos" (Duration in minutes), and "Ámbito" (Scope) with a dropdown menu showing "Seleccione uno". Below these is a date input field "Día" (Day) with the placeholder "dd/mm/aaaa". At the bottom is an orange "INGRESAR" button. A footer bar at the bottom right reads "Escuela Da Vinci 2021 — ADM".

The screenshot shows the same form as above, but with validation errors displayed in red boxes. The errors are: "El nombre del recorrido es obligatorio" (The route name is required), "La distancia es obligatoria" (The distance is required), "La duración es obligatoria" (The duration is required), "Debe ingresar un ámbito" (You must enter a scope), and "Debe ingresar una fecha válida" (You must enter a valid date). The "INGRESAR" button is still present.

The screenshot shows the same form again, but now with a green success message at the bottom: "Recorrido registrado" (Route registered). The rest of the interface is identical to the previous screenshots.

La función “guardar” (function) se ejecuta cuando se envía el form, en esa función tenemos nuestra propia validación de los datos en un “array de errores” cuando, mediante condicionales del lado de la lógica, “pusheamos” el error en caso de no ser lo que pedimos. Por ejemplo: Si se pone 0 en la distancia, se pushea en el array de errores “**La distancia debe ser mayor a 0**” y estará ahí hasta que se vuelva a mandar el formulario y hasta que el dato esté correcto.

Es una ayuda para el usuario y muy intuitivo ya que los errores están en rojo. Lo mismo en caso de mandar todos los datos correctamente, en verde figura “**Recorrido registrado**” con el mismo fin de ayudar al usuario.

Una vez registrado el recorrido, se guarda en **LocalStorage** un array que, stringificado con el key “data” nos va a dar los recorridos para mostrarlos en el componente y vista **“RECORRIDO.VUE”**

Este array se parsea y muestra cada vez que se “monta” vue. La idea es mantener todos los recorridos actualizados.

Los recorridos son recorridos por un **v-for** y cambia el estilo dependiendo si es en ciudad o montaña por medio de un operador ternario y un v-bind:class.

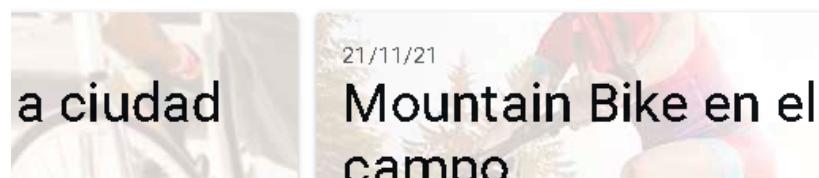
The screenshot shows a mobile application interface for 'Velobike'. At the top, there's a navigation bar with icons for Home, Ingresar (Login), and Recorridos (Routes). The 'Recorridos' icon is highlighted with a grey background. The main title 'Mis recorridos' is displayed above two cards. Each card represents a route:

- Tarde en la ciudad**: Date: 25/11/21. Type: Ciudad. Distance: 16 km. Duration: 180 minutos. Actions: BORRAR (Red button) and EDITAR (Orange button).
- Mountain Bike en el campo**: Date: 21/11/21. Type: Montaña. Distance: 25 km. Duration: 360 minutos. Actions: BORRAR (Red button) and EDITAR (Orange button).

At the bottom of the screen, a footer bar contains the text 'Escuela Da Vinci 2021 – ADM'.



OS



Dentro de la vista “**RECORRIDOS**” tenemos una función para **eliminar el recorrido**.

A través de un CONFIRM confirmamos con el usuario y se borra.

A cada recorrido le pasamos una **key** para que al borrar, sepa cuál recorrido borrar. Se borra del array que tiene todos los recorridos y **se actualiza el LocalStorage** en la misma función.

En caso de no tener recorridos, a través de un **v-if y v-else** mostramos que no hay recorridos aún y por medio de un <router-link> que lleva a la vista “ingresar” invitamos al usuario a registrar un nuevo recorrido

## Mis recorridos

No hay recorridos registrados todavía

+ INGRESAR

Dentro de recorridos también tenemos el botón “**EDITAR**” que pasa a través de parámetros dentro del **<router-link>**.

Pasamos todos los datos del recorrido: nombre, distancia, duracion, ambito, dia y adicional agregamos una **fecha** que se genera cuando el usuario INGRESA el recorrido, es único e irrepetible ya que contiene fecha con segundos, que usaremos **como ID del recorrido**.

Mandamos todos esos datos a “**EDITAR.VUE**” vista y componente, que muestra un formulario en su template que tiene todos los campos para editar.

Este formulario también tiene **validación** que no deja avanzar hasta que se cumpla todo lo requerido.

Una vez validado se crea un nuevo objeto con los nuevos datos y recorriendo nuestro array de recorridos (sacado de LocalStorage) se compara y si alguno de los recorridos coincide con el nuestro se borra con un **SPLICE**.

Esto hace que solo se borre el recorrido antiguo y lo **reemplazamos** con el nuevo recorrido por medio de un push, luego actualizamos el LocalStorage.

Si todo salió bien, se redirige a la vista “**RECORRIDO.VUE**” al usuario para poder ver los cambios.

## Editar recorrido

Nombre

Tarde en la ciudad

---

Distancia en km

16

---

Duración en minutos

180

---

Ámbito

Ciudad

---

Día

25/11/2021

---

**ACTUALIZAR**