

## Prueba - Cursos AlfaWeb

- Para realizar esta prueba debes haber estudiado previamente todo el material disponible en el LMS correspondiente al módulo.
- Una vez terminada la prueba, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el .zip en el LMS.
- Puntaje total: 10 puntos
- Desarrollo prueba: Individual

### Habilidades a evaluar

- Crear un entorno de desarrollo para Vue.js.
- Implementar Vue Router para proteger rutas y direccionar a un usuario.
- Crear rutinas que permitan conectarse con Firebase Authentication.
- Utilizar Vuex para gestionar la información suministrada por la base de datos.
- Crear rutinas que permitan conectarse con Cloud Firestore.
- Desarrollar las interfaces de la aplicación utilizando algún Framework de UI para Vue.
- Aplicar deploy a la aplicación final y subirla a Firebase Hosting.

### Descripción

La empresa de enseñanza en línea Cursos AlfaWeb se encuentra en la búsqueda de un programador en Vue.JS que implemente un pequeño sistema de administración de cursos en conjunto con Firebase para lograr administrar la base de datos y solo permitir el acceso a usuarios registrados. Igualmente necesita que incorpores algún Framework de UI para Vue que permita darle estilo al sistema.

Por lo tanto, la empresa necesita como primera fase:

- Crear un proyecto Vue que implemente Vue Router, Vuex y Firestore para gestionar los cursos de la empresa. Este proyecto debe estar enlazado con Firestore, por lo que debes trabajar con una cuenta en Google. Si no la posees, debes crear una.
- Crear una interfaz de acceso a usuarios mediante una vista de login (ver "Imagen 1") y otra de registro de usuarios (ver "Imagen 2"). Solo los usuarios registrados en el sistema de autenticación de Firestore podrán ingresar a la aplicación. La validación debe realizarse por dirección de correo electrónico y clave.
- Al validarse correctamente las credenciales de usuario en la interfaz de acceso "Login" o "Registro", se debe desplegar una interfaz principal "Home" (ver "Imagen 3")

donde se muestran todos los cursos que se encuentren en la base de datos mediante tarjetas o cards.

- Igualmente, debe existir un menú de navegación o “NavBar” que contenga el correo electrónico del usuario en la parte superior derecha en conjunto con un botón, icono o enlace, que al hacer clic cierre la sesión de usuario activa en ese momento, sin importar la vista que se encuentre activa. Al cerrar la sesión, el usuario será redirigido inmediatamente a la vista de acceso o “Login”.

Ahora, como segunda fase del proyecto, se debe:

- Crear una interfaz que permita administrar los cursos (ver “Imagen 4”). Esta interfaz solo debe activarse si el usuario se encuentra correctamente autenticado. Para acceder a ella se debe hacer mediante el menú de navegación “NavBar” o menú lateral con los nombres de las rutas disponibles.
- Asimismo, la interfaz de administración debe desplegar una tabla con el listado de cursos disponibles en la base de datos. Además debe tener botones o enlaces que permitan agregar, eliminar y/o editar cursos (ver “Imagen 4”).
- Por otra parte, en la interfaz de administración se deben mostrar diversos mensajes de carácter informativo, los cuales, deben indicar: (ver “Imagen 4”).
  - Cantidad total de alumnos permitidos.
  - Cantidad total de alumnos inscritos.
  - Cantidad total de cupos restantes.
  - Cantidad total de cursos terminados.
  - Cantidad total de cursos activos.
  - Cantidad total de cursos.
- El botón o enlace para agregar un nuevo curso ubicado en la interfaz de administración debe ejecutar una ventana emergente o modal al hacerle un clic. Esta ventana emergente o modal debe contener un formulario que permite agregar todos los datos (nombre, url de la imagen del curso, cupos, inscritos, duración, costo, código y descripción) (ver “Imagen 5”).
- Para agregar un nuevo curso se deben tomar en cuenta algunas características:
  - La cantidad de inscritos no debe ser mayor que la cantidad de cupos disponibles. Por lo tanto, se debe mostrar un error si esto ocurre y no se debe permitir agregar el curso.
  - El estado del curso (terminado) por defecto será falso (false).
  - La fecha de registro del curso será la que tome el sistema al momento de realizar el registro.
- Al agregar un curso a la base de datos de Firebase se debe desplegar un modal, popup o alert que nos pregunte si deseamos agregar dicho curso, el cual tendrá dos (02) botones que serán “Cancelar” y “Agregar Curso”. Si el usuario presiona

“Cancelar”, el modal deberá desaparecer. En cambio si el usuario presiona “Agregar Curso”, el sistema deberá agregar el registro en la base de datos de Firebase; al momento de agregar el registro, se deben actualizar todas las interfaces que muestran los cursos e informar que el curso fue agregado.

- Para eliminar cursos se debe desplegar una ventana emergente, modal, popup o alert que nos pregunte si realmente se desea eliminar el curso seleccionado, que tendrá dos (2) botones que serán “Cancelar” y “Sí, borrar”. Si el usuario hace un clic sobre “Cancelar”, el modal deberá desaparecer. En cambio si el usuario presiona “Sí, borrar”, el sistema deberá eliminar el curso en la base de datos de Firebase e indicar que el curso fue eliminado correctamente. Actualizando al instante todas las vistas que muestran información sobre los cursos.
- Para editar cursos, el usuario debe hacer un clic sobre el botón, enlace o icono relacionado al curso que desea editar mostrado en la interfaz de administración, para así redirigir al usuario a la interfaz de edición.
- En la interfaz de editar cursos (ver “Imagen 6”), se debe desplegar un formulario con la información precargada del curso que se desea editar, dando la oportunidad al usuario de editar todos los campos del curso:
  - Nombre (string)
  - URL de la imagen (string)
  - Cupos del curso (número)
  - Inscritos (número)
  - Costo (número)
  - Duración del curso (string)
  - Código (string)
  - Descripción (string)
  - Fecha de Registro (fecha)
  - Estado (booleano)
- La interfaz de edición tiene algunas características que se deben tomar en cuenta al momento de editar un curso:
  - La cantidad de inscritos no debe ser mayor que la cantidad de cupos disponibles, por lo tanto, se debe mostrar un error si esto ocurre y no se debe permitir la actualización del curso.
  - Si se modifica el estado del curso de activo a terminado (cerrado), la cantidad de inscritos debe automáticamente modificarse a cero.
- Esta interfaz de edición debe contar con un botón que permita volver a la interfaz de administración.

- Al actualizar el curso se debe desplegar un modal, popup o alert indicando que la actualización fue realizada, por lo tanto, el sistema deberá actualizar el curso correspondiente en la base de datos de Firebase modificando todas las vistas que muestran los cursos instantaneamente y retornando al usuario a la vista de administración automáticamente.

Con todo lo anterior listo, ya se puede compilar el proyecto en modo producción y enviarlo a Firebase Hosting.

Para el desarrollo de la interfaz de usuario, puede apoyarse en alguna librería como element-ui, bootstrap-vue, vuetify u otra que hayas visto a lo largo de las unidades anteriores.

Al finalizar, deberás entregar el código fuente de la aplicación desarrollada en un archivo comprimido. Además, debe compilar la aplicación en modo producción, desplegarla en Firebase Hosting y entregar en un archivo de texto (al interior del archivo comprimido) el enlace a su aplicación en Firebase junto con un usuario y clave para entrar al sistema. Este archivo se debe llamar "enlace-a-mi-firebaseapp.txt".







## Requerimientos

1. Crear el proyecto mediante el entorno de desarrollo de Vue-CLI, el cual debe contener: **(1 Punto)**
  - Por lo menos dos (02) dependencias externas instaladas y configuradas en el archivo main.js.
  - Vuex, Vue Router.
2. Utilizar Vue Router para controlar y proteger las rutas, así como crear los redireccionamientos necesarios. **(1 Punto)**
  - Implementar una ruta dinámica para la edición de los cursos a través de props, la cual permita identificar el curso a modificar, cargando la información existente en el formulario antes de ser editada.
  - Crear tres vistas protegidas mediante guardianes de navegación (administración, home y edición) y dos públicas (login y registro).
3. Crear rutinas que permitan conectarse con Firebase Authentication. **(2 Puntos)**
  - Implementar el método "create User With Email And Password" en la vista pública de registro para crear nuevos usuarios en Firebase Authentication.
  - Utilizar el método "sign In With Email And Password" en la vista de login para ingresar a la aplicación con datos de usuarios existentes.
  - Emplear el método "sign Out" en la barra de navegación o "navbar" para cerrar la sesión de un usuario activo en el momento.
  - Aplicar el método "on Auth State Changed" para identificar el ingreso o salida de usuarios de la aplicación, agregando una ventana modal o mensaje indicando que el usuario ingresó correctamente al sistema desde este método en específico.
4. Desarrollar el sitio web utilizando Vuex para almacenar y modificar los estados. **(1 Punto)**
  - Utilizar getters para dejar la información disponible a cualquiera de los componentes o vistas.
  - Implementar mutations para modificar los estados de la aplicación.
  - Emplear action para ejecutar las mutations en Vuex, traer la información de la base de datos de Firebase, así como para actualizar, agregar y eliminar cursos.
  - Utilizar los estados de Vuex para almacenar la información del usuario que inicie o se registre en la aplicación.

5. Crear rutinas que permitan conectarse con Cloud Firestore. **(2 Puntos)**
  - Implementar el método “on Snapshot” para leer la base de datos en tiempo real.
  - Utilizar el método “add” para agregar nuevos valores a la base de datos.
  - Emplear el método “update o set” para actualizar o modificar valores existentes en la base de datos.
  - Aplicar el método “delete” para eliminar valores existentes en la base de datos.
6. Desarrollar las interfaces de la aplicación utilizando algún Framework de UI para Vue. **(1.5 Puntos)**
  - Implementar al menos dos formularios en vistas diferentes.
  - Utilizar un menú de navegación o navbar para dejar disponibles las vistas.
  - Emplear al menos seis tarjetas o cards para mostrar información destacada.
  - Aplicar al menos una tabla para gestionar información.
7. Aplicar deploy a la aplicación final y subirla a Firebase Hosting. **(1.5 Puntos)**

## Recursos:

- Listado inicial de cursos a registrar en la base de datos:

Código	Nombre	Estado (Terminado)	Costo	Duración	Descripción	Cupos	Inscritos	Imagen
C00101	HTML Básico	true	10000	1 mes	Curso básico de HTML para principiantes.	35	0	
C00102	CSS para principiantes	false	10000	1 mes	Aprendiendo estilos con CSS desde el nivel más básico.	35	23	
C00201	JavaScript básico de 0 a 100	true	20000	2 meses	Programando para la web con JavaScript.	25	0	
C00202	JavaScript Avanzado	false	30000	2 meses	Curso con las nuevas actualizaciones de JavaScript.	20	10	
C00301	VueJS de 0 a 100	false	85500	5 meses	Framework Vue.js desde principiante a avanzado.	35	35	
C00302	Estilos con SASS	false	45000	1 mes	Mejorando los estilos con mayor potencia mediante SASS	40	35	

- Listado de direcciones URL para las posibles imágenes a utilizar:
  - ❖ HTML ([https://www.w3.org/html/logo/downloads/HTML5\\_Logo\\_512.png](https://www.w3.org/html/logo/downloads/HTML5_Logo_512.png))
  - ❖ CSS (<https://lineadecodigo.com/wp-content/uploads/2014/04/css.png>)
  - ❖ JavaScript Básico (<https://cuos.com.ar/img/tecnologias/js.png>)
  - ❖ JavaScript Avanzado ([https://i.blogs.es/545cf8/es6-logo/450\\_1000.png](https://i.blogs.es/545cf8/es6-logo/450_1000.png))
  - ❖ Vue ([https://thumbs.gfycat.com/PinkPiercingBull-size\\_restricted.gif](https://thumbs.gfycat.com/PinkPiercingBull-size_restricted.gif))
  - ❖ SASS ([https://miro.medium.com/max/512/1\\*9U1toerFxB8aiFRreLxEUO.png](https://miro.medium.com/max/512/1*9U1toerFxB8aiFRreLxEUO.png))

## Imágenes de referencia

The screenshot shows a web application header with a purple bar. On the left, there is a hamburger menu icon and the text 'Cursos AlfaWeb'. On the right, there is a user profile icon and the text 'USER2@USER.COM'. Below the header, the title 'Login de Usuario' is displayed. There are two input fields: 'E-mail' and 'Password'. The 'Password' field has a character count '0 / 20' on the right. Below the input fields, there are three buttons: 'INICIAR' (green), 'LIMPIAR FORMULARIO' (red), and 'LIMPIAR VALIDACIÓN' (orange).


Imagen 1. Interfaz de Login de Usuario  
Fuente: Desafío Latam.

The screenshot shows a web application header with a purple bar. On the left, there is a hamburger menu icon and the text 'Cursos AlfaWeb'. On the right, there is a user profile icon and the text 'USER2@USER.COM'. Below the header, the title 'Registro de Usuario' is displayed. There are two input fields: 'E-mail' and 'Password'. The 'Password' field has a character count '0 / 20' on the right. Below the input fields, there are three buttons: 'REGISTRAR' (green), 'LIMPIAR FORMULARIO' (red), and 'LIMPIAR VALIDACIÓN' (orange).

Imagen 2. Interfaz de Registro de Usuario  
Fuente: Desafío Latam.




## Lista de Cursos



### JavaScript Avanzado

**Información**


- Costo: **30000**  
Duración: 2 meses
- Cupos: **20**  
Completado: No
- Fecha de Registro:  
06-03-2021
- Descripción:  
Curso con las nuevas actualizaciones de JavaScript.



### CSS para principiantes

**Información**

- Costo: **10000**  
Duración: 1 mes
- Cupos: **35**  
Completado: No
- Fecha de Registro:  
05-03-2021
- Descripción:  
Aprendiendo estilos con CSS desde el nivel más básico.















### JavaScript básico de 0 a 100



**Información**


- Costo: **20000**  
Duración: 2 meses
- Cupos: **25**  
Completado: Si
- Fecha de Registro:  
05-03-2021
- Descripción:  
Programando para la web con JavaScript.

Imagen 3. Home o vista principal del sitio  
Fuente: Desafío Latam.


# Administración AGREGAR CURSO

Curso	Cupos	Inscritos	Duración	Costo	Terminado	Fecha	Acciones
JavaScript Avanzado	20	10	2 meses	\$30.000	No	06-03-2021	 
CSS para principiantes	35	23	1 mes	\$10.000	No	05-03-2021	 
JavaScript básico de 0 a 100	25	0	2 meses	\$20.000	Si	05-03-2021	 
VueJS de 0 a 100	35	35	5 meses	\$85.500	No	06-03-2021	 
Estilos con SASS	40	35	1 mes	\$4500	No	06-03-2021	 
HTML Básico	35	0	1 mes	\$10.000	Si	31-01-2021	 


Rows per page: 10 1-6 of 6  

 Cantidad total de alumnos permitidos: **190** alumnos

 Cantidad total de alumnos inscritos: **103** alumnos

 Cantidad total de cupos restantes: **87** alumnos

 Cantidad total de cursos terminados: **2** cursos.

 Cantidad total de cursos activos: **4** cursos.

 Cantidad total de cursos: **6** cursos.

Imagen 4. Interfaz de administración de cursos.  
Fuente: Desafío Latam.

The image shows a web application interface with a modal form titled "Agregarando Curso". The background is a dark purple header with a menu icon, the text "Cursos AlfaWeb", and a user profile "USER2@USER.COM" with an external link icon. The modal form is white and contains the following fields:

- Nombre:** A text input field with a character count "0 / 40".
- URL de la Imagen del curso:** A text input field.
- Cupos del curso:** A numeric input field with a value of "0" and up/down arrow controls.
- Inscritos en el curso:** A numeric input field with a value of "0" and up/down arrow controls.
- Duración del curso:** A text input field.
- Costo del curso:** A numeric input field with a value of "0" and up/down arrow controls.
- Código del curso:** A text input field.
- Descripción del curso:** A large text area for the course description.

At the bottom of the modal, there are three buttons: "AGREGAR" (green), "LIMPIAR FORMULARIO" (red), and "LIMPIAR VALIDACIÓN" (orange). The background application shows a list of courses on the left and a table of course details on the right, including columns for "Curso" and "Acciones".

Imagen 5. Ventana modal para agregar nuevos cursos.  
Fuente: Desafío Latam.



## Editando el curso: JavaScript Avanzado

Nombre

JavaScript Avanzado

19 / 20

URL de la Imagen del curso

[https://i.blogs.es/545cf8/es6-logo/450\\_1000.png](https://i.blogs.es/545cf8/es6-logo/450_1000.png)

Cupos del curso

20

Inscritos en el curso

10

Duración del curso

2 meses

Costo del curso

30000

Código del curso

C00202

Descripción del curso

Curso con las nuevas actualizaciones de JavaScript.

Fecha de Registro

06 - 03 - 2021



Terminado: No

ACTUALIZAR

LIMPIAR FORMULARIO

LIMPIAR VALIDACIÓN

REGRESAR

Imagen 6. Interfaz para editar un curso previamente seleccionado.  
Fuente: Desafío Latam.