**Prompt1**

Supón que eres un experto en Vue.js

---- ESTRUCTURA DEL PROYECTO VUE ----

La estructura de ficheros del proyecto es la siguiente:

frontend-enero-plantilla-2025

├── .gitignore

├── .vscode

│ ├── extensions.json

│ └── settings.json

├── index.html

├── jsconfig.json

├── package.json

├── public

│ └── favicon.ico

├── README.md

├── src

│ ├── App.vue

│ ├── assets

│ │ ├── base.css

│ │ ├── logo.svg

│ │ └── main.css

│ ├── components

│ │ ├── HelloWorld.vue

│ │ ├── icons

│ │ │ ├── IconCommunity.vue

│ │ │ ├── IconDocumentation.vue

│ │ │ ├── IconEcosystem.vue

│ │ │ ├── IconSupport.vue

│ │ │ └── IconTooling.vue

│ │ ├── TheWelcome.vue

│ │ └── WelcomeItem.vue

│ ├── firebase

│ │ ├── config

│ │ │ └── firebaseConfig.js

│ │ └── repositories

│ │ └── citasRepository.js

│ ├── main.js

│ ├── router

│ │ └── index.js

│ ├── stores

│ │ └── counter.js

│ └── views

│ ├── AboutView.vue

│ └── HomeView.vue

└── vite.config.js

---- TAREAS A REALIZAR ----

A partir de un proyecto Vue creado desde cero (con el comando npm create vue@latest):

-. Como backend se usarán los emuladores de Firebase

-. Hay un docker-compose.yml para poner en marcha los emuladores de Firebase en Docker.

debes de hacer lo siguiente.

-. Dentro del fichero src/firebase/repositories/citasRepository.js, debes de crear un método llamado “getCitaAleatoria()” que devuelva un documento de una colección llamada “Citas” al azar.

-. Debes de hacer que en el proyecto Vue el componente App contenga un único componente “Cita” que muestre texto y autor de una cita al azar, y un botón para obtener una nueva cita al azar.

---- IMPORTANTE ----

-. Dentro de Firestore de los emuladores de Firebase, ya está creada la colección “Citas”. Asimismo, ya existen varios documentos creados dentro de la colección “Citas”. TODOS estos documentos tienen un campo “texto” y un campo “autor”. Ambos campos son de tipo String.

-. Haz que el diseño visual de App.vue y de Citas.vue sea atractivo.

-. Para TODOS los estilos, DEBES de usar estilos CSS customizados (es decir, estilos que empiecen por ".", como ".container").

---- RESUMEN ----

-. A partir de toda la información que te he dado, adelante, haz tu trabajo ChadGPT.

-. Debes de darme TODO el código completo que generes NUEVO o que MODIFIQUES. A parte de darme el código completo, debes de indicarme EN TU RESPUESTA explícitamente aquellas partes de código NUEVAS que hayas creado o modificado.

-. Asimismo, DEBES de indicarme aquellos ficheros que NO necesiten ninguna modificación.

-. Si observas que te hace falta el código de algún otro fichero que no te haya adjuntado, DEBES de indicármelo claramente.

-. Además, debes de indicarme las rutas y los nombres de los nuevos ficheros creados y/o modificados.

---- FICHEROS ADJUNTADOS ----

Te adjunto los siguientes ficheros:

// docker-compose.yml

// main.js

// App.vue

// firebase/config/firebaseConfig.js

// firebase/repositories/citasRepository.js

**Prompt2**

Supón que eres un experto en JavaScript, CSS y HTML puros

---- ESTRUCTURA DEL PROYECTO VUE ----

La estructura de ficheros del proyecto es la siguiente:

js-estandar-enero-prueba-2025

├── .gitignore

├── index.html

├── package.json

├── public

│ └── vite.svg

└── src

├── counter.js

├── firebase

│ ├── config

│ │ └── firebaseConfig.js

│ └── repositories

│ └── citasRepository.js

├── javascript.svg

├── main.js

└── style.css

---- TAREAS A REALIZAR ----

A partir de un proyecto de Javascript puro sin frameworks creado con el comando:  
  
  
npm create vite@latest nombre-del-proyecto -- --template vanilla

Debes de tener en cuenta lo siguiente:

-. Como backend se usarán los emuladores de Firebase

-. Hay un docker-compose.yml para poner en marcha los emuladores de Firebase en Docker.

-. Debes de hacer lo siguiente.

-. Dentro del fichero src/firebase/repositories/citasRepository.js, debes de crear un método llamado “getCitaAleatoria()” que devuelva un documento de una colección llamada “Citas” al azar.

-. Debes de hacer que cada vez que se ingrese o se recargue la página, se muestre una cita al azar. Asimismo debes de hacer que cada vez que se pulse el botón se muestre de nuevo una cita al azar.

-. Debes de implementar esta funcionalidad únicamente con JS estándar, CSS y HTML puros. No debes de usar ningún framework JS.

---- IMPORTANTE ----

-. Dentro de Firestore de los emuladores de Firebase, ya está creada la colección “Citas”. Asimismo, ya existen varios documentos creados dentro de la colección “Citas”. TODOS estos documentos tienen un campo “texto” y un campo “autor”. Ambos campos son de tipo String.

-. Haz que el diseño visual de la vista sea visualmente atractivo.

-. Para TODOS los estilos, DEBES de usar estilos CSS customizados (es decir, estilos que empiecen por ".", como ".container").

-. Haz que a nivel visual se vea igual que en el caso de la funcionalidad que acabas de implementar con vue.

---- RESUMEN ----

-. A partir de toda la información que te he dado, adelante, haz tu trabajo ChadGPT.

-. Debes de darme TODO el código completo que generes NUEVO o que MODIFIQUES. A parte de darme el código completo, debes de indicarme EN TU RESPUESTA explícitamente aquellas partes de código NUEVAS que hayas creado o modificado.

-. Asimismo, DEBES de indicarme aquellos ficheros que NO necesiten ninguna modificación.

-. Si observas que te hace falta el código de algún otro fichero que no te haya adjuntado, DEBES de indicármelo claramente.

-. Además, debes de indicarme las rutas y los nombres de los nuevos ficheros creados y/o modificados.

---- FICHEROS ADJUNTADOS ----

Te adjunto los siguientes ficheros:

// docker-compose.yml

// src/main.js

// src/firebase/config/firebaseConfig.js

// src/firebase/repositories/citasRepository.js

// actualmente vacío

**Prompt 3**

Supón que eres un experto en Vue y Firebase.

---- ESTRUCTURA DEL PROYECTO ----

La estructura de ficheros actual del proyecto es la siguiente:

js-estandar-enero-prueba-2025

├── .gitignore

├── index.html

├── package.json

├── public

│ └── vite.svg

└── src

├── counter.js

├── firebase

│ ├── config

│ │ └── firebaseConfig.js

│ └── repositories

│ └── citasRepository.js

├── javascript.svg

├── main.js

└── style.css

---- TAREAS A REALIZAR ----

Debes de realizar lo siguiente:

-. Implementa en front y backend (backend dentro de la carpeta firebase/repositories) una funcionalidad en la que el usuario pueda puntuar una cita con un <input type = “number”> y un botón “puntuar” (entre 0 y 10).

-. Esta funcionalidad se debe de realizar sobre el mismo recuadro de mostrar una cita aleatoria.

-. Cada cita puede puntuarse múltiples veces.

-. En el recuadro de la cita correspondiente debe de mostrarse la media de puntuaciones que se han otorgado a dicha cita.

-. Para evitar tener que guardar las puntuaciones individuales puedes guardar en cada Cita simplemente la media hasta el momento y el número de puntuaciones. Para actualizar la media, aplica la fórmula: nuevaMedia = (mediaAnterior\*n+nuevoValor)/(n+1) donde n es el número de puntuaciones que había hasta el momento. Por ejemplo si la media hasta ahora era 5, había 2 puntuaciones y la nueva puntuación es 10, la nueva media será (5\*2+10)/3 = 6.67.

-. En firestore, ahora las citas tienen dos nuevos campo llamados "puntuacionMedia" y "numeroDePuntuaciones", ambos de tipo Number inicializados a 0 . Usa estos campos para la nueva lógica a implementar.

---- RESUMEN ----

-. A partir de toda la información que te he dado, adelante, haz tu trabajo ChadGPT.

-. Debes de darme TODO el código completo que generes NUEVO o que MODIFIQUES. A parte de darme el código completo, debes de indicarme EN TU RESPUESTA explícitamente aquellas partes de código NUEVAS que hayas creado o modificado.

-. Asimismo, DEBES de indicarme aquellos ficheros que NO necesiten ninguna modificación.

-. Si observas que te hace falta el código de algún otro fichero que no te haya adjuntado, DEBES de indicármelo claramente.

-. Además, debes de indicarme las rutas y los nombres de los nuevos ficheros creados y/o modificados.

Te adjunto los siguientes ficheros:

// docker-compose.yml

// main.js

// App.vue

// firebase/config/firebaseConfig.js

// firebase/repositories/citasRepository.js

**Prompt 4**

Supón que eres un experto en Vue y Firebase.

---- ESTRUCTURA DEL PROYECTO VUE ----

La estructura de ficheros del proyecto es la siguiente:

lista\_compra\_firebase\_vue

├── .gitignore

├── .vscode

│ └── extensions.json

├── README.md

├── index.html

├── jsconfig.json

├── package-lock.json

├── package.json

├── public

│ └── favicon.ico

├── src

│ ├── App.vue

│ ├── components

│ │ ├── About.vue

│ │ ├── AddItemForm.vue

│ │ ├── Contador.vue

│ │ ├── Item.vue

│ │ ├── Lista.vue

│ │ ├── ListaCompra.vue

│ │ └── LoginForm.vue

│ ├── config

│ │ └── firebaseConfig.js

│ ├── main.js

│ ├── repositories

│ │ ├── listaRepository.js

│ │ └── usuarioRepository.js

│ ├── router

│ │ └── index.js

│ └── services

│ └── authService.js

├── styles.css

└── vite.config.js

---- CONSIDERACIONES A TENER EN CUENTA ----

-. Como backend se usarán los emuladores de Firebase

-. Hay un docker-compose.yml para poner en marcha los emuladores de Firebase en Docker.

---- TAREAS A REALIZAR ----

-. Debes de agregar la funcionalidad para que un usuario pueda registrarse. En el formulario de registro, el usuario debe de introducir de forma obligatoria su username (Display name), su email y su contraseña (password).

-. Debes de modificar el código también para completar el CRUD entero de los ítems de la lista de la compra. Ahora cuando un usuario inicie sesión, dicho usuario podrá ver todos los ítems, crear un nuevo item, borrar un item, editar el nombre de un item y ver los detalles de un item.

-. Cuando se pulse sobre el botón de editar, el texto del ítem debería cambiar por un campo de texto y el botón de editar debe de cambiar su texto por “Guardar”.

---- IMPORTANTE ----

-. La lógica del registro dentro del backend debe de repartirse de forma apropiada entre los archivos repositories/listaRepository.js y services/AuthService.js para respetar el principio de separación de responsabilidades.

-. La lógica del CRUD de ítems de la lista de la compra debe de manejarse íntegramente dentro del fichero repositories/listaRepository.js.

-. Las modificaciones que hagas en el frontend deben de estar lo más modularizadas posibles, es decir, dichas modificaciones deben de repartirse entre todos los siguientes ficheros: ListaCompra.vue, Lista.vue, AddItemForm.vue, Item.vue.

---- RESUMEN ----

-. A partir de toda la información que te he dado, adelante, haz tu trabajo ChadGPT.

-. Debes de darme TODO el código completo que generes NUEVO o que MODIFIQUES. A parte de darme el código completo, debes de indicarme EN TU RESPUESTA explícitamente aquellas partes de código NUEVAS que hayas creado o modificado.

-. Asimismo, DEBES de indicarme aquellos ficheros que NO necesiten ninguna modificación.

-. Si observas que te hace falta el código de algún otro fichero que no te haya adjuntado, DEBES de indicármelo claramente.

-. Además, debes de indicarme las rutas y los nombres de los nuevos ficheros creados y/o modificados.

---- FICHEROS ADJUNTADOS ----

Te adjunto los siguientes ficheros:

// repositories/listaRepository.js

// repositories/usuarioRepository.js

// services/authService.js

// config/firebaseConfig.js

// docker-compose.yml

// App.vue

// router/index.js

// main.js

// components/LoginForm.vue

// components/ListaCompra.vue

// components/AddItemForm.vue

// components/Lista.vue

// components/Item.vue

// components/About.vue

// components/Contador.vue

// styles.css