

La prueba consiste en realizar un pequeño desarrollo a partir de un mock up y una breve definición funcional.

Tecnologías librerías a usar:

- Vue 3
- Vite
- Pinia
- TailwindCSS

Puedes utilizar herramientas adicionales, y se valorará su uso.

La entrega deberá consistir en el código fuente e instrucciones para su ejecución (para tu mayor comodidad, puedes subirlo a tu perfil de Github y pasarnos el enlace al proyecto). Si has utilizado herramientas adicionales, por favor, incluye en la entrega cuáles son y cómo las has utilizado (puedes describir estos detalles en el README.md del proyecto).

Diseño

Comentarios

VERSIÓN 1
[Título del archivo a subir 2.pdf](#)
Documento / PDF (3.8 MB) 28 dic 2018 / 14:45

Buenas tardes, acabo de subir el primer archivo con el primer caso práctico, espero revisión.
28 dic 2018 / 14:45

iGran trabajo Silvia! Pero creo que deberías revisar el punto 3. Intenta añadir un poco más de contexto y un apartado final de conclusiones personales.
29 dic 2018 / 10:11

VERSIÓN 2
[Título del archivo a subir 2.pdf](#)
Documento / PDF (4.8 MB) 30 dic 2018 / 8:22

Realizados los cambios según feedback.
30 dic 2018 / 8:22

Perfecto, excelente trabajo.
30 dic 2018 / 15:25

Descripción

El desarrollo consiste en la implementación de una ventana modal en la que el usuario, como estudiante, mantiene una conversación con su tutor. Los mensajes de ambos se muestran en orden cronológico inverso (mensajes más recientes abajo). Intercalados con los mensajes se muestran las entregas que el estudiante ha hecho a su tutor. Cada entrega consiste en un número de versión, nombre, tipo y tamaño de fichero, fecha de entrega y una URL de descarga.

Desde esta modal, el estudiante debe poder escribir y enviar mensajes. La modal deberá estar precargada con mensajes de estudiante y tutor, y varias entregas.

Valoración

Se valorará:

- Facilidad a la hora de ejecutar el desarrollo.
- Correcto funcionamiento.
- Claridad y legibilidad del código.
- La descomposición en componentes del diseño.
- Conocimiento de Vue.
- Cobertura de tests de componentes y funcionalidades creadas
- Consideraciones sobre accesibilidad.
- Documentación de todas las suposiciones realizadas.

No se valorará:

- Pequeñas diferencias con respecto al diseño.
- Errores debidos a diferencias sutiles en la interpretación de los requisitos funcionales.
- Rendimiento o tamaño del código resultante.

No es necesaria la comunicación con ninguna API o servicio. No es necesario que las URL de descarga de la entrega sean reales.