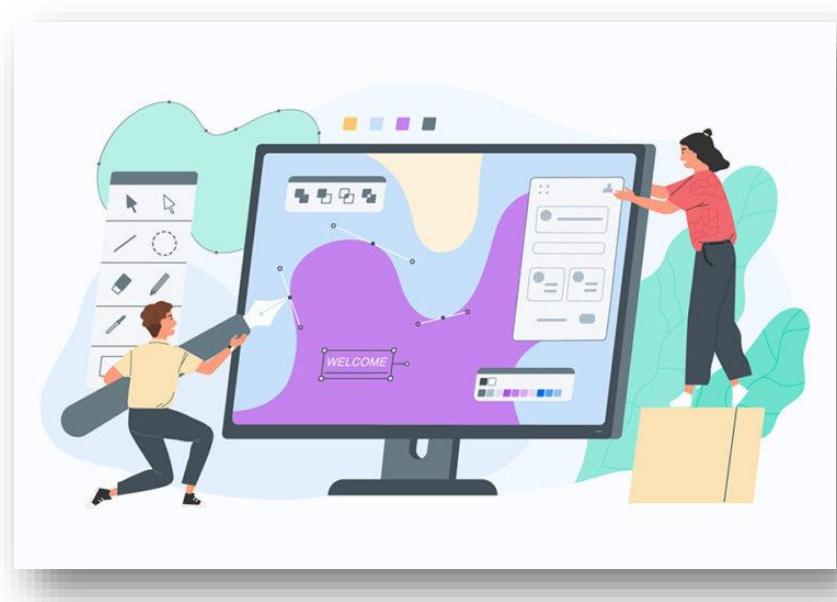


Visualización de los datos de una API rest o web service, en un diseño frontend con framework Vue3



Mostrar en una SPA (Single Page Application), los datos servidos (por GET) desde una API Rest (endpoints en formato JSON). Posibilidad de gestionar un servidor Fake (Json-Server) y realizar operaciones completas CRUD que añadan un elemento de valor a nuestra aplicación. Utilizar un agente de IA, en la gestión del proyecto de forma expresa y documentada

Crear una interactividad en la interface, para poder realizar en el frontend una gestión de datos.

html

Css

Responsive design

Javascript

Git

JSON

SCRUM

API Rest

Linux

UX/UI

SEO

VueJS

Trello/Jira

Marcos de competencias

Desarrollador web y web móvil

Recurso(s)

HTML - CSS - JavaScript

Contexto del proyecto

Dentro de un proyecto conjunto para un cliente, desarrollamos una propuesta de valor para unos usuarios que consuman servicios de información especializado en los temas PRODUCTOS FINANCIEROS, TIEMPO METEOROLÓGICO, INFORMACIÓN DE DEPORTES y MANGA-ANIME

Se asigna por el docente, a cada grupo un área determinada. Dentro de esta asignación cada equipo debe investigar un ámbito concreto para ofrecer una propuesta de valor que deberá ser validada por el formador

La propuesta de valor deberá cumplir, al menos los siguientes requisitos

- Utilizar una o varias APIs públicas, que permitan su utilización. Comprobación de la disponibilidad de su uso (con registro, api key, autenticación por token, etc).
- Ofrecer una funcionalidad que genere un valor reconocible para un conjunto de usuarios (establecer un *buyer-persona*), que motive la realización de la aplicación.

Nuestra labor consiste en crear un frontend para gestionar los contenidos de un elemento temático. Para la visualización de la petición de datos que nos llega por la API, utilizaremos para la capa de interface un framework de frontend, que en este caso va a ser Vue. En este segundo caso las condiciones son similares en cuanto a replicar el modo de visualización de los datos. Pero adaptados a las estrategias de desarrollo del framework.

Como elemento añadido, es posible realizar la gestión de la conexión de datos a un servidor propio con una API de datos propios (como Json Server) que simule la utilización de un backend. Este servidor va a atender las peticiones de un CRUD completo (Get, Post, Put y Delete) integrando esta capa de visualización en el proyecto frontend, combinado con la recepción de datos desde una API externa

Las funcionalidades descritas constituyen una etapa en el proceso de proyecto completo. Está en la lógica de un desarrollo completo, un frontend unido a una capa de gestión de datos. El ámbito de aplicación podría ser un blog, una valoración de usuarios, etc.

Modalidades pedagógicas

Secuencia pedagógica

Trabajo en grupos de 4, los grupos son autónomos para la realización del proyecto. En cada grupo, designamos la figura de Scrum Master El equipo trabaja con una metodología Agile, se organiza una reunión cada 2 días con el product owner y cliente (representado por el formador). Al final del proyecto, el equipo realiza una demostración de 45 minutos al cliente (representado por el capacitador) de las características agregadas.

Herramientas

Framework de desarrollo frontend Vue3-Vite (composition api) Editor de código Visual Studio Code, Control de versiones Git, Figma, Json-Server Internet. Otras herramientas son también posibles (Bootstrap, Sass, Tailwind)

Criterios de rendimiento

El material propio de desarrollo, como el código fuente ha sido entregado en modo y plazo correcto

Se han utilizado aplicaciones de control de versiones tal y como atestigua el repositorio remoto facilitado al efecto. Se trabaja con distintas ramas (branch) de Git Hub. Se utilizan diferentes ramas (producción-main, developer, las propias de cada coder o tarea)

Los scripts de código desarrollados funcionan de forma eficiente, sin bugs y sin problemas de renderización. El naming (los términos que identifican los elementos) de variables, constantes, funciones, parámetros, comentarios, etc será en inglés. El código esta formateado, comentado en inglés y adecuadamente tabulado

Utilizar técnicas de naming en código como BEM, es un criterio altamente recomendado

La petición a los datos servidor por una instancia remota es adecuada y estable

Los elementos de datos que se envían por medio un endpoint (o varios) de un servidor local se visualizan en el frontend

Los comportamientos de selección, mostrar y ocultar de los ítems de datos funcionan de forma adecuada

La estructura de los elementos web y su apariencia es adecuada y refleja los criterios de accesibilidad y legibilidad.

La demostración es acertada. De forma concisa y clara. Se ha respondido a la demanda de información adicional de forma adecuada

La actitud y las habilidades de comunicación en la exposición de la tarea realizada, son propios del estándar profesional en el ámbito del desarrollo en programación y a los usos habituales en contactos profesionales.

Trello inicialmente y luego Jira será la aplicación online colaborativa para gestionar las líneas de proceso del sprint con las metodologías SCRUM

Requerimientos técnicos

Desarrollo Avanzado con vue3/vite como framework para el frontend

- Se trabaja con Vue 3, con la configuración de vue/vite. Se podrán utilizar las modalidades acreditadas como composition api.
- Se crearán diversas views o pages que den un carácter dinámico y completo al website a través de diversos contenidos accesibles a los usuarios.
- Se utilizará la modularización creando componentes para poder albergar los datos de visualización en un módulo que se pueda replicar en bucle (una lista desordenada, o unos section dentro de un article).
- Los componentes aplican la filosofía del Atomic Web Design
- Ni las vistas, ni los componentes llevan los nombres de I@s coders del Development Team. Se utilizan criterios semánticos o funcionales (se llaman expresando lo que son o lo que hacen). Y lo mismo en los elementos de código se nombran en inglés
- Se utilizarán las directivas que permiten dar comportamientos integrados al desarrollar, cada cual en su modus operandi: v-bind para renderizar datos, v-show para mostrar y ocultar, v-if para condicionales o v-for para las iteraciones.
- Se utilizarán archivos router para marcar las rutas, los props para pasar información entre los módulos parent y child y los eventos emit para hacer lo correspondiente en sentido inverso. O las técnicas de provide-injection, para pasar datos. O la utilización de Pinia para controlar el estado
- El tratamiento de la información asíncrona, utilizará recursos como las etiquetas <suspense/>, gestionando los estados de conexión y espera
- Se utilizará la reactividad de datos primarios y de objetos (ref, reactive, toRef, y otras funcionalidades)

- El proyecto será totalmente responsive con al menos dos breakpoints (diseño para mobile, tablet y desktop)
- Bocetado completo del webSite para los tres ámbitos del dispositivo. Con versión en blanco y negro y en color
- Se considera una mejora la utilización de varias APIs para proporcionar al usuario un aporte más completo del servicio especializado
- El proyecto deberá contar con test a través de una librería de testing como Jest, Vuetest, Cypress,... Debe haber al menos 5 test que midan diferentes contextos de ejecución.
- Es un requisito incluir en el código la utilización al menos de un composable y funciones del ciclo de vida del framework

Se considera mejora la utilización de librerías para el manejo de los estados (Pinia) o el uso de Typescript

Se considera mejora realizar el Build de los archivos finales de producción y realizar el deployment, en un servidor remoto gratuito (Netlify o similar).

[Utilización de agente de IA](#)

En este proyecto se anima a la utilización de forma medida y organizada de agente de IA, y en este contexto de forma específica se detalla un criterio de rendimiento

Modalidades de evaluación

Se debe hacer Daily todos los días y se revisará. El sprint review tendrá:

Entrega del código fuente, Presentación del WebSite con proyector mostrando el resultado en formato usuario web y código

El sprint review tendrá: Evaluación mediante la presentación del WebSite con proyector mostrando el resultado en formato usuario web y código.

La presentación contendrá una parte inicial en inglés (leído o comentado) de 10 minutos, con participación de todo el equipo de desarrollo

Tras la presentación se realizará la Sprint retro utilizando el modelo Kanban

Entregables

Código fuente, en forma de repositorio de Git disponible en GitHub

Bocetado completo del prototipado realizado en Figma (formato pdf o imágenes montadas en pdf)

Realizar una presentación para asegurar la continuidad y coherencia del sprint review. La presentación contendrá los bocetos de FIGMA y la propuesta de valor inicialmente presentada que haya recibido validación

Enlace a la herramienta de colaboración utilizada (Trello-Jira)

Trabajo para ser entregado en DOS semanas.