**Proyecto Integrador (E-Commerce) — Frontend**

***Requerimientos (Sprint 3)***

Luego de haber realizado las entregas en las iteraciones anteriores, el cliente quiere empezar a generar un mayor dinamismo en su página y una mayor interacción del usuario.

Para ello, solicita incorporar Javascript al proyecto y poder comenzar a trabajar en las funcionalidades de su web.

Para la primera parte, el cliente requiere que podamos empezar a generar el carrito de compras dinámico que requiere un control de los productos que se venden en su plataforma.

Recuerden que ya realizamos en la entrega anterior el front end del carrito y, en esta oportunidad debemos continuar con la funcionalidad para que se realice la funcionalidad correctamente.

Deben utilizar el local storage para almacenar la lista de productos que el usuario vaya seleccionando (a través de los eventos de js que consideren necesarios para agregar un producto al carrito, siendo el carrito un array [ ] de productos).

También debemos generar una función que calcule el precio total de los productos agregados al carrito para mostrar el total en el detalle de compra (luego de haber seleccionado los productos a comprar y listarlos en la página del carrito).

Recuerden que también pueden agregar botones para eliminar productos del carrito que quieran descartar. (Aunque no va a ser obligatorio en este Sprint).

Por otra parte, dado que generamos un listado de productos, debemos cargar esos productos desde una base que los contenga. (Cuando lleguemos al módulo de Backend podremos profundizar y migrar los datos a las bases de datos, por el momento nos manejaremos con el storage). Luego de tener los productos cargados en un formato que nos permita manipularlos y poder iterarlos dinámicamente para generar nuestras cards de productos, usaremos las cards de bootstrap para completar los datos de cada producto de forma dinámica con Javascript.

También, visto que el usuario nos pidió tener datos válidos para que puedan ser almacenados en una base de datos, se nos pide que validemos los formularios de registro de usuario y login del mismo.

Una vez que el usuario complete correctamente todos los campos y se registre, deberá poder iniciar sesión con ese usuario creado (para eso, por el momento, utilizaremos el local storage como vimos en el ejercicio de práctica de las clase del 3 de mayo).

Para poder iniciar sesión, los datos de usuario y contraseña ( o email y contraseña) deberán validarse mediante el almacenamiento de esos datos del registro en el localstorage, que serán consultados para realizar el logueo del usuario a la plataforma (pueden dejar un mensaje de bienvenida al usuario una vez inició sesión con un alert, pero no es de carácter obligatorio).

El cliente nos anticipó que nos pedirá más funcionalidades luego, así que debemos utilizar buenas prácticas de programación para tener un código ordenado, prolijo, comentado (para saber qué realiza cada función), nombre de variables, constantes y funciones descriptivas al objeto u comportamiento que va a realizar.