**COVID-19 acelera el comercio electrónico y pagos sin contacto en América Latina.**

La propagación del nuevo coronavirus trae consigo desafíos en todos los niveles; desde la capacidad de respuesta de los sistemas de salud y el involucramiento social para acompañar las disposiciones, hasta el alcance de las estrategias para proteger a los más vulnerables y mitigar los efectos negativos en la economía. Así mismo, diversos grupos de ciudadanos proactivos se organizan para aportar soluciones a las problemáticas más urgentes del contexto actual. Si bien es tiempo de incertidumbre y solidaridad, también lo es de ingenio y creatividad. Por ello, buscamos responder de forma innovadora a los desafíos que nos trae la pandemia [3]. En nuestro caso nos enfocaremos en aportar soluciones para mejorar la economía para lo cual se ha decidido aportar al COMERCIO ELECTRONICO y a los PAGOS SIN CONTACTO.

Nuestro proyecto integrador pretende desarrollar un sistema de comercio electrónico. Sabemos que en estos tiempos de pandemia todos los comercios deben migrar a la modalidad en línea por lo que para este proyecto puedes elegir la venta de cualquier conjunto de productos (dispositivos electrónicos, libros, artículos de primera necesidad, artículos perecederos, etc.).

Los requerimientos mínimos del sistema están divididos en 2 partes:

1. Funcionales
2. De codificación (programación)

Los requisitos funcionales de nuestro sistema de venta de productos son

1. Listar todos los productos disponibles de tu tiendita
   1. Hay más de un producto (al menos 10 productos)
2. Agregar productos para su compra (similar a un carrito de compras). No puedes quitar productos una vez que lo has seleccionado
3. Mostrar el total en pesos de los productos agregados para su compra junto con el nombre de los productos
4. Opción para comprar los productos seleccionados
   1. Sólo es un mensaje que indica que se han comprado los artículos

Los requisitos de codificación (programación) son:

1. Los productos deben ser clases (las clases con atributos “iguales” deberán utilizar herencia)
2. Utiliza al menos una relación de composición, asociación, agregación y herencia
3. Agrega los setters y getters correspondientes a cada clase
4. Todas tus clases deben tener al menos un constructor por default y un constructor con parámetros
5. Debe haber al menos una instancia de todas tus clases. Prueba dicha instancia con alguno de los métodos que incluya