

Momento Valorativo #5

Programación Distribuida

Juan Carlos **Zuluaga**

Condiciones

- Se evaluará todo lo que hemos visto en desarrollo de dispositivos móviles con énfasis en MVVM.
- El valor de este trabajo es del 20% que depende de la sustentación individual, más que del producto entregado.
- Debe trabajar este proyecto de manera **individual**, llevando un registro detallado y progresivo de cada uno de los COMMITS que realice.
- La aplicación se construirá en Xamarin.Forms pero se probará en Android. Es decir, no se evaluará la plataforma iOS o Windows Phone.
- Fecha de entrega, la segunda clase de la semana del 14 de noviembre (última semana de clases).
- Debe usar una de las combinaciones aprobadas por Adobe Color (<https://color.adobe.com/es/explore/most-popular/?time=all>) seleccione una de su agrado y combine los colores para que la aplicación quede agradable para usar.

Enunciado

Se requiere construir una aplicación que permita a una comunidad de usuarios hacer chequeos de precios de manera colaborativa. Es decir, si deseo adquirir un producto X, podré consultar en este aplicativo, en que competidores venden este producto y donde sale más favorable comprarlo. Los usuarios podrán matricular competidores, productos, y precios de estos productos en los diferentes competidores. El aplicativo debe tener:

- Una opción de login que pida correo y contraseña. Con posibilidad de recordarme en ese dispositivo, registrarme como nuevo usuario y recordar la contraseña en caso que el usuario la olvide (se mandará un correo con la nueva contraseña).
- El usuario podrá modificar sus datos como nombres, apellidos, correo, contraseña y foto de perfil. Deberá poder usar la misma aplicación para tomarse la foto y establecerla como foto de perfil.
- El usuario deberá poder matricular competidores (por ejemplo: Éxito Colombia, Jumbo las Vegas, Euro Florida, ETC). Estos competidores deben contener como mínimo: Nombre, Dirección, Teléfono, Horario, Longitud y Latitud, los cuales podrán ser cargados mediante el GPS del teléfono. Se debe registrar cual usuario matriculó el competidor.
- Se debe validar que no se pueda ingresar 2 veces el mismo competidor.
- También se podrán modificar datos del competidor y eliminarlo.
- El usuario deberá poder matricular productos (por ejemplo: Café Sello Rojo x 450 G, Leche Colanta x 1 Lt, ETC). Estos productos deben contener mínimo: Nombre, Precio y código de

barras. El código de barras debe poder capturarse con la cámara del teléfono usándola como un lector de código de barras. Se debe registrar cual usuario matriculó el producto.

- Se debe validar que no se pueda ingresar 2 veces el mismo producto, es decir, 2 productos con la misma descripción.
- También se podrán modificar datos del producto y eliminarlo.
- El usuario podrá matricular precios de un artículo en un competidor (por ejemplo: la leche Colanta en el Éxito de Colombia vale \$2,000 y en Euro Florida vale \$1,900). Al matricular el precio debe matricular la fecha en que hizo el registro del precio y debe quedar registro de que usuario registro que precio.
- Para cualquier operación donde involucre el producto, lo deberá poder buscar por código de barras usando la cámara del teléfono.
- El usuario podrá consultar para un producto en cuales competidores lo encuentra, ordenando los precios del más barato al más caro y en el mapa debe mostrar la ubicación actual y la ubicación de los competidores donde encuentre dicho producto.
- El aplicativo debe poder trabajar sin conexión a internet, es decir, al tener conexión debe cargar toda la data necesaria para trabajar y almacenarla en una base de datos local SQLite. Cuando cree nuevos registros los debe marcar como registros “no subidos” y cada que haga una operación de grabado o consulta de información, debe “subir” los registros al servicio API y marcar los registros como sincronizados.

Condiciones técnicas

- Debe usar todo lo que vimos en el ejemplo de MVVM, hasta el manejo de estilos de botones y los templates de las list view en el diccionario de recursos.
- Debe usar SQLite para la persistencia de los datos y las tablas deben estar relacionadas.
- Debes de construir tu base de datos del backend y del front end para cumplir con los requerimientos.
- Debes construir el API de comunicaciones y publicarlo en un Hosting para poderlo consumir desde los teléfonos (recomiendo, gratuito: <https://somee.com> y de pago: <https://www.negox.com>).
- Todo el código debe ir en inglés, incluye los comentarios y los commits, todo el interfaz de usuario debe ir en español.
- Debe refrescar bien las listas y manejar de forma correcta los eventos de notificación al cambio de propiedades.

¡Mucha suerte!