

EVALUACIÓN	Obligatorio	GRUPO	TODOS	FECHA	JULIO 2021
MATERIA	Taller para Desarrollo de Dispositivos Móviles				
CARRERA	Analista en Tecnologías de la Información / Analista Programador / Programador Web				
CONDICIONES	<p>- <u>Fecha de Entrega</u>: 10/ 08 /21 hasta las 21:00 hs.</p> <p>- <u>Puntos</u>: Máximo: 100 Mínimo: 1</p> <p>LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40MB EN FORMATO ZIP, RAR O PDF.</p> <p>IMPORTANTE</p> <ul style="list-style-type: none"> - Los grupos deben estar conformados por hasta un máximo de dos personas del mismo dictado. - Inscribirse - Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO" 				

Introducción

El restaurante más famoso de Montevideo "Pinta Rico", desea implementar su propio sistema de pedidos. A diferencia de los clientes que van a comer al local, quienes realicen pedidos para retirar, podrán solicitar únicamente comida que el local ya tiene pre-elaborada (empanadas, brochettes, pizzetas, croquetas, etc.).

El sistema consta de un backend, una API y una aplicación mobile.

El backend y la API ya fueron desarrollados, mientras que para el desarrollo de la aplicación, han decidido contratarnos a nosotros.

Por tratarse de un producto algo complejo, inicialmente avanzaremos sobre una primera etapa, la cual consta de crear un prototipo para que sea validado por el cliente antes de comenzar con la segunda etapa del proyecto.

Este prototipo se debe entregar funcionando para dispositivos Android, y debe permitir el registro de los clientes, visualizar los productos disponibles y realizar pedidos sin tener en cuenta la parte del pago online.

No tendremos acceso al backend previamente desarrollado, pero sí nos han disponibilizado toda la API y una documentación técnica de la misma, la cual podemos encontrar en <https://app.swaggerhub.com/apis-docs/gabrielarce/pinta-rico/1.0.0>.

Detalle de funcionalidades

Registro

Para que un usuario utilice la aplicación debe haber iniciado sesión y para eso, se debe haber podido registrar con anterioridad.

Para efectuar el registro, se debe ingresar: correo electrónico, *password*, nombre, apellido y dirección (nombre de calle y número de puerta). Todos los datos son obligatorios, la contraseña debe tener al menos 8 caracteres y la dirección de correo debe tener un formato válido (*@*. *). Además,

si bien no debemos controlarlo, sabemos que no pueden existir dos usuarios con igual correo electrónico.

Toda la información ingresada debe enviarse al servidor a través de la operación correspondiente de la API. Si bien la API controlará que todas las condiciones antes mencionadas se cumplan, se debe validar toda la información a nivel de la aplicación y, sin necesidad de realizar una llamada a la API, se deberá mostrar al usuario un error descriptivo.

Inicio de sesión

Como se dijo anteriormente, nuestra app debe tener una pantalla de “log in” para que el usuario se identifique. En esa pantalla el usuario deberá ingresar obligatoriamente correo y *password* para luego autenticarse utilizando la operación de la API correspondiente. Al igual que con el registro, se deben validar los datos ingresados para evitar comunicarse con la API sabiendo que la información ingresada no es válida.

Una vez que el usuario se identifica en la aplicación, si cierra la app sin haber solicitado el cierre de sesión, al volver a ingresar debe ser “recordado” por el dispositivo, lo que significa que sin tener que realizar ninguna acción, el usuario estará *logueado*. Esto se puede realizar o bien volviendo a iniciar sesión de forma automática (habiendo guardado correo y password), o bien utilizando la operación de autenticación por token que ofrece la API.

Pantalla principal

Una vez que se inicie sesión (independientemente de si es de forma automática o por solicitud del usuario), se deberá visualizar la pantalla principal (*home*).

La información que se deberá presentar en esta pantalla queda a criterio del desarrollador, pero se debe tener en cuenta que, al ser la pantalla principal, sería bueno que sea atractiva a la vista del usuario.

Se requiere que, de algún modo, se pueda acceder al menú de opciones de la aplicación. El mismo puede ser un *side menu* o un conjunto de *tabs* inferiores (cada *tab* sería una opción del menú). Las pantallas a las que se debe poder acceder a través del menú son: “Productos”, “Favoritos” y “Mis Pedidos”.

Pantalla “Productos”

Debemos mostrar todos los productos, los cuales serán obtenidos a través de la operación de la API correspondiente. De cada producto se debe mostrar: nombre, precio, foto, código, etiquetas y estado (si está en stock o no). Además, en cada producto deberá aparecer un ícono que permita marcarlo como favorito (o sacarlo de favoritos en caso de que ya lo esté).

En la parte superior de la pantalla, antes del listado, debe haber un campo de texto que nos permita filtrar los productos por nombre (*substring*) o por contener una etiqueta. El filtro por nombre se debe realizar a través de la API, la cual en la operación de consulta de productos permite incluir un parámetro para indicar el criterio por el cual se quiere filtrar. Por otro lado, el filtro por etiquetas debe ser implementado en la propia aplicación.

La pantalla deberá contener también un botón que nos permita filtrar un producto realizando un escaneo de su código QR.

Para poder probar esta funcionalidad, el cliente nos ha disponibilizado un sitio web donde se encuentran todos los códigos QR de los productos.

Este sitio se encuentra en: <https://ort-tallermoviles.herokuapp.com/>.

El código QR contiene la URL de consulta a la API para obtener solamente ese producto a partir de su código.

Al presionar sobre un ítem del listado, el usuario deberá navegar hacia el detalle del producto.

Detalle de un producto

En esta nueva pantalla, debemos mostrar todos los datos del producto seleccionado. La información a mostrar se deberá obtener utilizando la operación de la API correspondiente. Ahora podremos saber y mostrarle al usuario tanto la descripción como el puntaje.

En caso de que exista stock del producto, debemos habilitar un botón para realizar un pedido.

Alta de pedido

En esta pantalla, del producto se deberá mostrar el título, la foto y el precio unitario. Además, se debe permitir al usuario ingresar la cantidad deseada (que debemos asegurarnos de que sea un número Natural mayor a cero). A medida que el usuario modifica la cantidad deseada, se deberá actualizar el importe total del pedido (que se debe estar mostrando al usuario).

Visto que Pinta Rico tiene varias sucursales, es necesario que el cliente indique en qué sucursal retirará su pedido, lo cual deberá hacer a través de un combo desplegable.

Es fundamental que, al momento de elegir la sucursal, el usuario pueda ver en un mapa su ubicación (tomada automáticamente desde su dispositivo, NO se debe tener en cuenta para esto la dirección ingresada al momento del registro) y la ubicación de cada una de las sucursales, indicando nombre y distancia (entre la sucursal y la ubicación del usuario).

Pantalla “Mis Pedidos”

Se deben listar todos los pedidos realizados por el usuario. La información a mostrar deberá ser la misma que en el listado de productos, agregando el nombre de la sucursal de retiro, el precio total (en lugar del precio del producto) y el estado del pedido.

En caso de que el pedido aún se encuentre en estado pendiente, se deberá permitir al usuario que ingresar un comentario a modo de calificación. Para esto, el cliente propone que al presionar el pedido aparezca un *modal dialog* con un campo de texto y un botón de confirmación, pero está abierto a la interfaz de usuario que el equipo de desarrollo proponga. Cualquiera sea la visual por la que se opte, una vez ingresado el comentario, debemos enviarlo a través de la API, la cual se encargará de marcar el pedido como entregado. Luego de guardado el comentario, se deberá refrescar el listado de los pedidos para mostrar la información actualizada.

Pantalla “Favoritos”

Aquí debemos listar todos los productos que el usuario guardó en favoritos y debemos permitir que sean desmarcados. Los productos marcados como favoritos no se podrán guardar mediante la API, por lo que tendremos que usar el almacenamiento interno de la app para resolverlo. Tengamos en cuenta que más de un usuario podría estar usando el mismo dispositivo.

Cerrar sesión

¡No podemos olvidarnos de esto!

La app debería permitirle al usuario realizar un cierre de sesión, conservando los favoritos, pero obligando a realizar el *login* nuevamente.

Requerimientos extra


- Para el diseño, se deberá utilizar el *framework Onsen UI*.
- Para almacenamiento local, se utilizará *localStorage*.
- La aplicación deberá ser generada sólo para dispositivos Android, utilizando el framework cordova (para lo cual se dispone de una guía de instalación y configuración).
- Se deberán controlar todos los tipos de error devueltos por la API.
- Las fotos de los productos se obtendrán desde el servidor de la API, formando su URL de la siguiente forma:
[https://ort-tallermoviles.herokuapp.com/assets/imgs/\\${producto.urlImagen}.jpg](https://ort-tallermoviles.herokuapp.com/assets/imgs/${producto.urlImagen}.jpg).
- La aplicación deberá controlar en todo momento que el dispositivo tenga acceso a internet. En caso de que se pierda se deberá indicar al usuario mediante una alerta.
- Al realizar el *login* contra la API, si la operación es exitosa, nos retornará un token de acceso el cual será utilizado para todas las operaciones que exijan el parámetro x-auth según la documentación de la API.

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

- **Obligatorios**

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. Ingresá al sistema de Gestión.
2. En el menú, seleccioná el ítem "Evaluaciones" y la instancia de evaluación correspondiente, que figura bajo el título "Inscripto".
3. Para iniciar la entrega hacé clic en el ícono: 
4. Ingresá el número de estudiante de cada uno de los integrantes y hacé clic en "Agregar". El sistema confirmará que los integrantes estén inscriptos al obligatorio y, de ser así, mostrará el nombre y la fotografía de cada uno de ellos. Una vez agregados todos los integrantes, hacé clic en "Crear equipo".

Cualquier integrante podrá:

- **Modificar la integración del equipo.**
- **Subir el archivo de la entrega.**

5. Seleccioná el archivo que deseás entregar. Verificá el nombre del archivo que aparecerá en la pantalla y hacé clic en "Subir" para iniciar la entrega. Cada equipo (hasta 3 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar). El archivo a subir debe tener **un tamaño máximo de 40mb**

Quando el archivo quede subido, se mostrará el nombre generado por el sistema (1), el tamaño y la fecha en que fue subido.

6. El sistema enviará un e-mail a todos los integrantes del equipo informando los detalles del archivo entregado y confirmando que la entrega fue realizada correctamente.
7. Podés cerrar la pestaña de entrega y continuar utilizando Gestión o salir del sistema.
8. **La hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
9. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc.).
10. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador o Coordinación adjunta **antes de las 20:00hs.** del día de la entrega, a través de los mails gervaz@ort.edu.uy, alamon@ort.edu.uy y terra@ort.edu.uy, o telefónicamente al 29021505 - int 1156 (de 8:00 a 14:00 hs) y 1436 (de 17:30 a 20:00 hs).

Si tuvieras una situación particular de fuerza mayor, debes contactarte con suficiente antelación al plazo de entrega, al Coordinador de Cursos (gervaz@ort.edu.uy) o Secretario Docente (paulos@ort.edu.uy).