

Trabajo Práctico: Marketplace de Servicios con Insumos

Stack:

- **Mobile:** React Native (Expo o CLI).
- **Web (admin/portal):** React.js (Vite o CRA).
- **Auth:** hardcodeada (lista fija de usuarios con `rol`).
- **Persistencia:** por ahora en **estado local** o JSON mock (pueden usar `json-server` o un store global).
- **Estado:** React Context + Reducer.

Objetivo

Construir una plataforma donde un **Solicitante** publique servicios (ej.: *limpieza de jardín y pileta*), **Proveedores de Servicio** coticen esos pedidos, y **Proveedores de Insumos** ofrezcan los insumos requeridos para ejecutar el servicio. El Solicitante debe **comparar cotizaciones** (precio total, plazos, etc.) y **seleccionar** una.

Roles y Permisos

- **Solicitante**
 - Pública servicios (con título, descripción, ubicación, fecha deseada, categorías y lista de insumos requeridos).
 - Recibe cotizaciones de Proveedores de Servicio.
 - Visualiza ofertas de insumos asociadas a su servicio.
 - Compara cotizaciones (por precio, tiempo estimado, rating promedio del proveedor) y **selecciona una**.
 - Cambia estado del servicio: *Publicado* → *En evaluación* → *Asignado* → *Completado*.

- **Proveedor de Servicio**

- Ve servicios publicados (filtro por categoría/ubicación/fecha).
- Envía **cotización**: precio, plazo, notas, ítems incluidos/excluidos.
- Puede editar o retirar su cotización mientras el servicio esté en *Publicado/En evaluación*.

- **Proveedor de Insumos**

- Publica insumos (nombre, categoría, unidad, precio unitario, stock, tiempos de entrega).
- Ofrece **packs** de insumos para un servicio concreto (bundle con precio total).
- Ve demanda de insumos (qué insumos piden los servicios publicados) y propone equivalencias.

Autenticación hardcodeda: en el login se elige usuario de una lista (o email/clave fija) y se inyecta en el contexto `{id, nombre, rol}`.

Requisitos Funcionales (mínimos)

1. **Login** simple (hardcoded) con selección de rol → redirección a su panel.
2. **Publicación de Servicio (Solicitante)**
 - Form con: título, descripción, dirección/ciudad, fecha preferida, categoría (ej. jardinería/piscinas), insumos requeridos (lista: nombre, cantidad/unidad).
 - Estado inicial: *Publicado*.
3. **Listado y detalle de servicios**
 - Todos (según permisos) pueden ver detalle de un servicio: datos, insumos requeridos, cotizaciones recibidas y ofertas de insumos.
4. **Cotizaciones (Proveedor de Servicio)**
 - Crear/editar/eliminar cotización: `precio, plazoDias, detalle`.
5. **Ofertas de Insumos (Proveedor de Insumos)**

- Crear packs de insumos para un servicio o publicar catálogo general.

6. Comparador (Solicitante)

- Tabla/tarjetas comparativas: precio total, plazo, rating proveedor (mock), desglose opcional.
- Acción **“Seleccionar cotización”** → cambia servicio a *Asignado* y bloquea nuevas cotizaciones.

7. Historial del Solicitante

- Servicios creados con estado y cotización seleccionada (si aplica).

8. Validaciones básicas

- Campos obligatorios, precios > 0, fecha válida, stock suficiente en packs.

9. UX

- Búsqueda/filtrado por categoría/ciudad/fecha.
- Feedback de acciones (toasts/snackbars).
- Estados vacíos (no hay cotizaciones, no hay insumos, etc.).

Entregables

- **Web (React.js):** Portal principal (puede ser panel para todos los roles o solo back-office/administración).
- **Mobile (React Native):** App orientada al rol Solicitante y/o Proveedor (a elección: mínimo 2 roles con pantallas dedicadas).
- **README** con:
 - Instrucciones para correr web y mobile.
 - Usuarios hardcodeados (email/clave/rol).
 - Decisiones técnicas (estado, routing, cómo mockean datos).

- **Demo** (gif/videos cortos) mostrando: publicar servicio → recibir cotización → comparar → seleccionar.
-

Rutas / Pantallas sugeridas

Web (React.js)

- `/login`
- `/` (dashboard según rol)
- `/servicios` (listado)
- `/servicios/:id` (detalle + cotizaciones + ofertas de insumos)
- `/insumos` (catálogo/ABM, rol Insumos)
- `/cotizaciones` (mis cotizaciones, rol Proveedor Servicio)

Mobile (React Native)

- **Tabs** (según rol): *Servicios, Publicar, Cotizaciones/Ofertas, Perfil*.
 - **Stacks**: Detalle de servicio, Form de Cotización, Form de Pack de Insumos.
-

Modelo de Datos (sugerido)

```
// Usuarios hardcodeados
User { id: string; nombre: string; email: string; rol: "SOLICITANTE"
| "PROVEEDOR_SERVICIO" | "PROVEEDOR_INSUMOS" }

// Servicio
Service {
  id: string;
  solicitanteId: string;
  titulo: string;
  descripcion: string;
  categoria: "jardineria" | "piscinas" | "limpieza" | "otros";
```

```
    direccion: string;
    ciudad: string;
    fechaPreferida: string; // ISO
    insumosRequeridos: { nombre: string; cantidad: number; unidad:
string }[];
    estado: "PUBLICADO" | "EN_EVALUACION" | "ASIGNADO" | "COMPLETADO";
    cotizacionSeleccionadaId?: string;
    createdAt: string;
}
```

// Cotización (proveedor de servicio)

```
Quote {
    id: string;
    serviceId: string;
    proveedorId: string;
    precio: number;
    plazoDias: number;
    detalle?: string;
    ratingProveedorMock?: number; // 1-5
    createdAt: string;
}
```

// Insumo (catálogo)

```
Supply {
    id: string;
    vendedorId: string;
    nombre: string;
    categoria: string;
    unidad: string; // ej. "lts", "kg", "unidad"
    precioUnit: number;
    stock: number;
}
```

// Pack de insumos ofrecido a un servicio

```
SupplyOffer {
    id: string;
    serviceId: string;
    vendedorId: string;
    items: { supplyId: string; cantidad: number }[];
    precioTotal: number; // calculado
    notas?: string;
}
```

```
    createdAt: string;  
}
```

Flujo clave: comparación y selección

1. Solicitante abre **Comparador** en el detalle del servicio.
 2. Ve **cotizaciones** (precio/plazo/rating) + **packs de insumos** (precio/stock).
 3. Puede:
 - Ordenar por **menor precio** o **menor plazo**.
 - Ver detalle expandible.
 4. Clic en “**Seleccionar cotización**” → servicio a *ASIGNADO* y persiste `cotizacionSeleccionadaId`.
 5. (Opcional) Marcar como **COMPLETADO** y dejar *rating*.
-

Reglas y Consideraciones

- **Auth hardcodeada:**
 - Array `USERS` en un módulo común (web y mobile) o duplicado simple.
 - Login “simulado”: si `email` y `password` coinciden con un registro, setear `currentUser` en contexto y `navigate` al dashboard.
- **Sin backend:**
 - Guardar datos en estado global; permitir recargar con un seed inicial.
 - **Opcional:** `json-server` para CRUD local y persistencia simple.
- **Control de permisos:**
 - Render condicional por `rol`.

- Proveedor solo cotiza, no edita servicio. Insumos solo publica packs/catálogo.
 - **UX básica pero clara:**
 - Formularios con validación mínima.
 - Mensajes de éxito/error.
 - Loading y empty states.
-

Criterios de Aceptación (MVP)

- Login hardcodeado por rol.
 - El Solicitante **crea** servicios con lista de insumos requeridos.
 - El Proveedor de Servicio **envía cotizaciones** a un servicio.
 - El Proveedor de Insumos **ofrece packs** de insumos para un servicio.
 - El Solicitante **compara** cotizaciones y **selecciona** una.
 - Estados del servicio actualizados correctamente.
-

Evaluación

- **Código funcional**
- **Defensa oral del código y entendimiento del código desarrollado**