





**✅ Te propongo un dashboard de bienvenida claro y funcional, con este menú o pantalla inicial:**

**🧭 Menú principal del admin (al entrar)**

1. **📅 Citas del día (resumen rápido)**
   * Lista de las próximas citas de hoy.
   * Estado: pendiente, confirmada, realizada, cancelada.
   * Con botón rápido para confirmarlas o gestionarlas.
2. **🚨 Alertas o pendientes**
   * 🔔 “Tienes 3 citas sin confirmar”
   * 🔔 “2 cancelaciones recientes”
   * Botón para ver el detalle de esas citas.
3. **👥 Gestión de personal**
   * Acceso a dentistas: ver, añadir, desactivar.
   * Acceso a usuarios (si tienes roles distintos).
4. **🗓️ Agenda general**
   * Calendario mensual/semanal de todas las citas.
   * Filtro por dentista, estado, etc.
5. **🧾 Pacientes**
   * Lista de pacientes registrados por cita.
   * Posibilidad de buscar, ver historial de citas o modificar datos.
6. **➕ Crear nueva cita**
   * Acceso directo a la pantalla de crear cita desde cero.
7. **⚙️ Configuración (opcional)**
   * Horarios disponibles.
   * Configurar tratamientos.
   * Roles y accesos.

**💡 Diseño del dashboard**

Visual, con tarjetas, colores para estados (verde = confirmada, rojo = cancelada, etc), y accesos rápidos.

Ejemplo:

| **🔔 Pendientes de confirmar** | **📅 Citas hoy** | **👥 Dentistas activos** |
| --- | --- | --- |
| 3 citas | 7 citas | 4 |

**🎯 ¿Qué esperaría yo como admin?**

* **Entrar y en 5 segundos saber qué tengo que hacer**.
* Confirmar citas rápido.
* Ver cambios recientes.
* Buscar información sin perder tiempo.

### 📌 Resumen visual ideal:

[ Pendientes: 3 ] [ Hoy confirmadas: 8 ] [ Libres hoy: 5 ] [ Ver calendario ]

────────── Miércoles, 28 de Mayo ──────────

09:00 | Dra. García | Juan Pérez | Confirmada

09:30 | Dra. García | — LIBRE — |

10:00 | Dr. López | María Gómez | Pendiente

Dentistas:

[ García ] [ López ] [ Ramírez ] ▼

**✅ 1. Mostrar reservas pendientes**

**🔹 Objetivo:** Ver cuántas reservas están en estado "pendiente" y listarlas.

**Qué necesitas:**

* Un **servicio Angular** (ReservaService) que haga un GET al backend tipo:

bash

CopiarEditar

GET /reservas?estado=pendiente

* Un **componente** tipo ReservasPendientesComponent que:
  + Consuma ese servicio.
  + Muestre una lista/tarjetas.
  + Quizás un badge con el total.

**Herramientas Angular:**

* HttpClientModule (para llamadas al backend)
* ngFor (para recorrer las reservas)
* ngIf (para mostrar “No hay pendientes”)
* Pipe async si usas Observables

**✅ 2. Mostrar agenda de hoy**

**🔹 Objetivo:** Ver reservas confirmadas (y libres) de hoy, por hora.

**Qué necesitas:**

* En el backend: una ruta tipo:

bash

CopiarEditar

GET /reservas?fecha=2025-05-28

o un endpoint tipo /agenda/hoy

* En el front:
  + Componente AgendaHoyComponent
  + Recorrer las horas (ideal: backend te dé los bloques de tiempo)
  + Mostrar:
    - Hora
    - Nombre del paciente (si hay)
    - Dentista asignado
    - Estado (confirmada, libre, cancelada)

**Herramientas Angular:**

* ngFor
* Pipes personalizados (opcional, para formato de fecha/hora)
* Un pequeño servicio AgendaService

**✅ 3. Navegación entre vista diaria / semanal / mensual**

**🔹 Objetivo:** Poder cambiar el día/semana/mes sin recargar todo.

**Qué necesitas:**

* Un componente tipo AgendaSelectorComponent:
  + Selector de fecha (puede ser input type="date" o librería como ngx-datepicker)
  + Selector de vista (Día, Semana, Mes)

**Herramientas Angular:**

* @angular/forms
* ngModel o ReactiveForms
* @Input/@Output para comunicar fecha seleccionada a AgendaHoyComponent

**📌 Recomendado orden de implementación:**

| **Paso** | **Qué hacer** | **Duración estimada** |
| --- | --- | --- |
| 1 | Crear ReservaService con métodos GET | 15-30 min |
| 2 | Crear ReservasPendientesComponent | 30-60 min |
| 3 | Crear AgendaHoyComponent con datos de hoy | 1-2 h |
| 4 | Añadir selector de fecha/vista (Día/Semana) | 1 h |
| 5 | Refinar con estilo, badges, botones | 1-2 h |

**Qué podrías hacer ahora para el backend:**

* **CRUD de citas**: crear, listar (por paciente, profesional, fechas), actualizar estado, cancelar, etc.
* **Asignación de tratamientos a citas**: que puedas actualizar la cita con el tratamiento correspondiente.
* **Control de horarios y duración**: validar que no se solapen citas del mismo profesional.
* **Filtrado y búsqueda**: que puedas consultar citas por paciente, profesional, fecha o estado.

**¿Por qué enfocarte en esto?**

* Es lo que mueve la app, y ya tienes usuarios para asignar.
* Al controlar citas bien, el front puede mostrar calendario, historial y permitir acciones claras.
* Te servirá para conectar luego con tratamientos y notificaciones.

¡Perfecto! Gracias por el contexto real. Si la **secretaria** va a gestionar esto y no es técnica, entonces **NO debería generar disponibilidad manualmente** con scripts ni JSON.

**✅ Lo ideal es que el backend lo haga automáticamente por ella.**

**🧠 ¿Qué deberías hacer como desarrollador?**

Implementar una función en tu backend que:

1. ✅ Cada semana (o cuando lo indique la secretaria desde el panel)…
2. 🔁 Genere las availabilities para cada profesional
3. Solo para los días laborales (lunes a viernes)
4. Y solo si **no existen ya** registros para esa fecha

**🛠️ ¿Cómo lo gestionaría la secretaria?**

Desde un panel de administración del frontend:

**💻 Interfaz:**

Un botón que diga:

"Generar disponibilidad semanal"

O un campo como:

**[ Fecha de inicio ] [Generar disponibilidad ⏭️]**

Al hacer clic:

* Se envía una petición a tu API backend.
* El backend genera las entradas en la tabla availabilities automáticamente.

**📦 ¿Qué se guarda en la base de datos?**

En la tabla availabilities:

| **professional\_id** | **date** | **slot\_id** | **status** |
| --- | --- | --- | --- |
| 7 | 2025-06-03 | 1 | libre |
| 7 | 2025-06-03 | 2 | libre |
| 8 | 2025-06-03 | 1 | libre |