GENERAR LAS HORAS DEL CALENDARIO POR DOCTOR

Para poder “armar” un calendario semanal en el que, para cada doctor, tengas sus huecos de 1 h marcados como **disponible** / **reservado** / **no disponible** teniendo en cuenta tanto la tabla de disponibilidades como la de bloqueos, una forma típica en MySQL 8 sería:

**1. Generar los slots horarios (09–14 y 15–18)**

Con un **CTE recursivo** creamos los bloques de 1 h:

sql

CopiarEditar

WITH RECURSIVE hrs AS (

SELECT TIME('09:00:00') AS slot\_start

UNION ALL

SELECT ADDTIME(slot\_start, '01:00:00')

FROM hrs

WHERE slot\_start < '17:00:00'

),

working\_slots AS (

SELECT slot\_start,

ADDTIME(slot\_start, '01:00:00') AS slot\_end

FROM hrs

WHERE slot\_start BETWEEN '09:00:00' AND '13:00:00'

OR slot\_start BETWEEN '15:00:00' AND '17:00:00'

)

**2. Para una semana y un profesional, ensamblamos la disponibilidad**

sql

CopiarEditar

, week\_days AS (

-- Ajusta '2025-05-19' por el lunes de la semana que quieras

SELECT DATE('2025-05-19') + INTERVAL n DAY AS dt

FROM (SELECT 0 AS n UNION ALL SELECT 1 UNION ALL SELECT 2

UNION ALL SELECT 3 UNION ALL SELECT 4) AS days

)

SELECT

p.id\_disponibilidad AS availability\_id,

w.dt AS date\_availability,

s.slot\_start AS start\_time,

s.slot\_end AS end\_time,

CASE

WHEN b.id\_bloqueo IS NOT NULL THEN 'no disponible'

WHEN d.estado\_disponibilidad = 'reservado' THEN 'reservado'

ELSE 'disponible'

END AS status

FROM working\_slots AS s

CROSS JOIN week\_days AS w

LEFT JOIN disponibilidades\_profesionales AS d

ON d.professional\_id = @PROF\_ID

AND d.fecha\_disponibilidad = w.dt

AND d.hora\_inicio\_disponibilidad = s.slot\_start

AND d.hora\_fin\_disponibilidad = s.slot\_end

LEFT JOIN bloqueos AS b

ON b.professional\_id = @PROF\_ID

AND b.tipo\_bloqueo = 'profesional'

AND b.fecha = w.dt

AND NOT (

s.slot\_end <= b.hora\_inicio

OR s.slot\_start >= b.hora\_fin

)

ORDER BY w.dt, s.slot\_start;

**¿Qué hace este query?**

1. **working\_slots**: genera todos los tramos de 1 h entre 09–14 y 15–18.
2. **week\_days**: lista los cinco días de lunes a viernes de la semana deseada (ajusta la fecha de inicio).
3. **CROSS JOIN** slots × días → obtienes todas las casillas posibles.
4. **LEFT JOIN** con disponibilidades\_profesionales para ver si ese hueco existe y si ya está marcado como “reservado”.
5. **LEFT JOIN** con bloqueos (filtrando solapamientos) para marcar “no disponible” cuando hay bloqueo activo.
6. El CASE final da prioridad a **bloqueos** (no disponible), luego a **reservas**, y por defecto **disponible**.

**3. Cómo usarlo en tu servicio**

* Pasa @PROF\_ID (ID del doctor) y la fecha del lunes de la semana.
* El resultado lo mapeas a tu modelo TS:

ts

CopiarEditar

export interface WeeklySlot {

dateAvailability: string; // e.g. "2025-05-19"

startTimeAvailability: string; // "09:00:00"

endTimeAvailability: string; // "10:00:00"

statusAvailability: 'disponible'|'reservado'|'no disponible';

}

* Agrúpalo por fecha en el cliente (o directamente en el SQL usando JSON\_ARRAYAGG) para generar la estructura:

ts

CopiarEditar

interface ProfessionalWeek {

nameProfessional: string;

weeklyAvailability: { date: string; slots: WeeklySlot[] }[];

}

Con este enfoque tienes siempre la **malla completa** de huecos de lunes a viernes y el sistema te señala automáticamente en qué casillas cae un bloqueo o una reserva.