

Bases de datos

Trabajo Práctico Especial 2018

Grupo: 15

Alumnas:

Gonzalez Fernanda - fgonzalez@alumnos.exa.unicen.edu.ar

Torrez Dara - dsamantatorrez@gmail.com

Indice:

1. [Dominio del sistema de reservas](#)
2. [Ajuste del esquema](#)
3. [Elaboración de restricciones](#)
4. [Servicios](#)
5. [Vistas](#)
6. [UI del sitio](#)

Introducción

El trabajo actual se centró en la modificación del esquema brindado por la cátedra sobre reservas de departamentos, para ajustarlo de la mejor manera al dominio del problema. Y se añadieron aclaraciones para cada uno de los cambios requeridos.

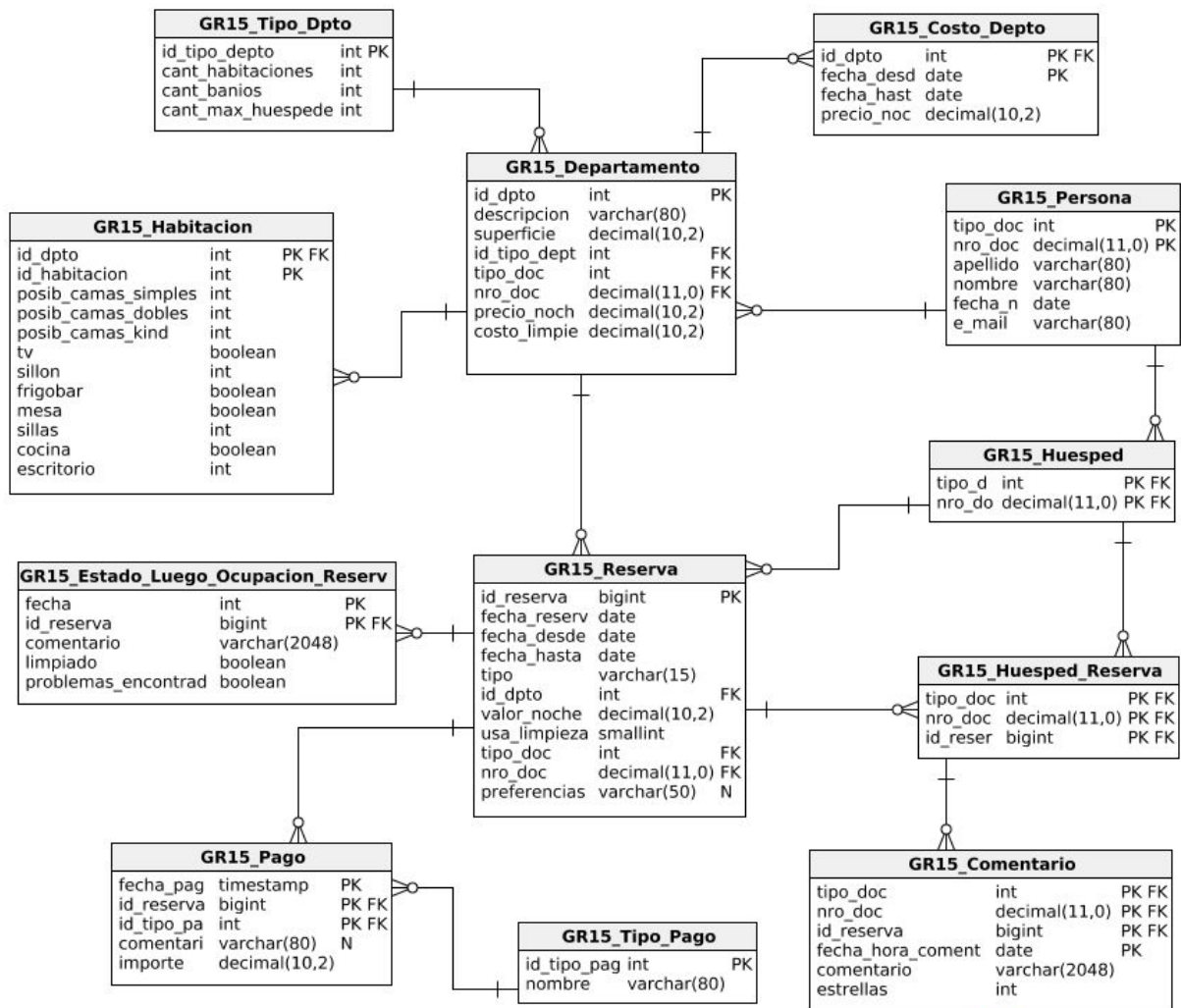
1. Dominio del sistema de reservas

- Hay dueños de casas o departamentos que los proponen para alquiler temporario.
 - Tabla Persona cumple con este requerimiento (Dueño). Se podría considerar casa como un tipo de depto.
- La administración consiste en el control y seguimiento completo de las reservas, cada una debe tener el tipo de departamento, fechas, preferencias, etc.
 - Tabla Reserva necesita los siguientes campos: preferencias (suponemos un campo de texto)
- El periodo mínimo que se puede reservar un departamento es un día.
 - Un check para los campos : fecha_desde, fecha_hasta.
- En la tabla Departamento se agrega la columna 'ciudad' ya que va a ser utilizada varias veces en este trabajo.
- El sistema debe llevar control del mantenimiento del departamento (es decir si fue limpiado y/o si se encontró algún problema en el mismo)
 - Agregar a Estado_luego_de_reserva un campo: limpiado (boolean), problemas_encontrados (boolean) deberían estar detallados en los comentarios.
- Las habitaciones de los departamentos pueden cambiar su configuración de una reserva a otra, respecto de la cantidad o tipo de camas y/o comodidades (televisor, sillones, frigobar, escritorios, mesas, etc).
 - Se debería agregar "escritorios", la tabla Habitacion cumple con el requerimiento.

2. Ajuste del esquema

Se consideró que el campo e_mail en la tabla Persona, tiene que ser único.

Se agregó un check para validar que el periodo mínimo que se pueda reservar un departamento sea de un día. Se agregó la columna ciudad a Departamento que va a ser útil para un servicio.



3.Elaboración de restricciones

- a. Que las fechas de las reservas sean consistentes, es decir que la fecha de inicio de la reserva sea menor que la fecha de finalización.
 - i. Interpretación: Esto se había analizado antes desde la lectura del dominio del problema (El periodo mínimo que se puede reservar un departamento es un día) por eso se agregó en el archivo creación el constraint CK_GR15_DURACION_RESERVA .
 - ii. También contemplamos que la fecha_reserva fuera consistente: fuera menor que fecha_desde. Se actualizó la tabla reserva con el constraint CK_GR15_CONSISTENCIA_FECHA_RESERVA

Tabla/s	Atributo/s	Chequeo	Tipo de restricción
Reserva	Fecha_hasta, Fecha_desde fecha_reserva	<p>CONSTRAINT CK_GR15_DURACION_RESERVA CHECK ((fecha_hasta - fecha_desde) >= 1) NOT DEFERRABLE INITIALLY IMMEDIATE,</p> <p>Luego se actualizó la tabla para añadir este constraint:</p> <p>ALTER TABLE GR15_RESERVA ADD CONSTRAINT CK_GR15_CONSISTENCIA_FECHA_RESERVA CHECK (fecha_desde >= fecha_reserva)</p>	Fila

Prueba	Resultado esperado
<pre>INSERT INTO gr15_reserva(id_reserva, fecha_reserva, fecha_desde, fecha_hasta, tipo, id_dpto, valor_noche, usa_limpieza, tipo_doc, nro_doc, preferencias) VALUES (6, '2017-10-05', '2017-10-09', '2017-10-07', 1, 1, 100, 1, 2, 5,null);</pre>	<p>ERROR: new row for relation "gr15_reserva" violates check constraint "ck_gr15_duracion_reserva"</p>
<pre>INSERT INTO gr15_reserva(id_reserva, fecha_reserva, fecha_desde, fecha_hasta, tipo, id_dpto, valor_noche, usa_limpieza, tipo_doc, nro_doc, preferencias) VALUES (6, '2017-10-06', '2017-10-05', '2017-10-07', 1, 1, 100, 1, 2, 5,null);</pre>	<p>ERROR: new row for relation "gr15_reserva" violates check constraint "ck_gr15_consistencia_fecha_reserva"</p>

- b. Que el detalle de las habitaciones sea consistente con el tipo de departamento, es decir que si el tipo de departamento es de 2 habitaciones, en el detalle se consideren como máximo 2 habitaciones.
- i. **Interpretación:** Si se agrega una habitación para un departamento, no puede ser mayor a la cantidad que está establecido en tipo de departamento.

Tabla/s	Atributo/s	Chequeo	Tipo de restricción
Tipo_Dto Departamento Habitacion	cant_habitaciones id_dpto id_tipo_depto	<pre>CREATE ASSERTION AS_GR15_HABITACION_MAX_CANTIDAD_HABITACIONES CHECK NOT EXISTS (SELECT d.id_dpto, t.cant_habitaciones,COUNT(*) FROM GR15_DEPARTAMENTO d INNER JOIN GR15_TIPO_DPTO t ON t.id_tipo_depto = d.id_tipo_depto INNER JOIN GR15_HABITACION h ON d.id_dpto = h.id_dpto GROUP BY d.id_dpto, t.cant_habitaciones</pre>	Base de datos

		HAVING t.cant_habitaciones < COUNT(*))	
--	--	--	--

Prueba	Resultado esperado
INSERT INTO gr15_habitacion(id_dpto, id_habitacion, posib_camias_simples, posib_camias_dobles, posib_camias_kind, tv, sillón, frigobar, mesa, sillas, cocina, escritorio) VALUES (1, 8, 1, 2, 1, true, 1, false, true, 1, false, 1);	ERROR: No puede haber más habitaciones de las creadas para este tipo de departamento
INSERT INTO gr15_habitacion(id_dpto, id_habitacion, posib_camias_simples, posib_camias_dobles, posib_camias_kind, tv, sillón, frigobar, mesa, sillas, cocina, escritorio) VALUES (2, 6, 1, 2, 1, true, 1, false, true, 1, false, 1);	INSERT 0 1

- c. Que tanto la persona que realiza la reserva como los huéspedes no sea el propietario del departamento

Tabla/s	Atributo/s	Chequeo	Tipo de restricción
Departamento Huesped _reserva Reserva	Tipo_doc Nro_doc Id_reserva id_depto	CREATE ASSERTION AS_GR15_RESERVA_VALIDAR_QUE_NO_SEA_PROPIETARIO CHECK NOT EXISTS (SELECT 1 FROM GR15_DEPARTAMENTO d INNER JOIN GR15_RESERVA r ON r.id_dpto = d.id_dpto INNER JOIN GR15_HUESPED_RESERVA h ON r.id_reserva = h.id_reserva WHERE (d.nro_doc = r.nro_doc AND d.tipo_doc = r.tipo_doc) OR (d.nro_doc = h.nro_doc AND d.tipo_doc = h.tipo_doc))	Base de datos

Prueba	Resultado Esperado
INSERT INTO gr15_reserva(id_reserva, fecha_reserva, fecha_desde, fecha_hasta, tipo,	ERROR: El propietario del departamento no

id_dpto, valor_noche, usa_limpieza, tipo_doc, nro_doc, preferencias) VALUES (10, '2017-10-05', '2017-10-05', '2017-10-07', 1, 1, 100, 1, 1, 1,null);	puede reservarlo
UPDATE gr15_reserva SET tipo_doc=1, nro_doc=1 WHERE id_reserva=1;	ERROR: El propietario del departamento no puede reservarlo
UPDATE gr15_huesped_reserva SET tipo_doc=1, nro_doc=1 WHERE id_reserva=1;	ERROR El propietario del departamento no puede ser huésped de la reserva
INSERT INTO gr15_huesped_reserva(tipo_doc, nro_doc, id_reserva) VALUES (1, 2, 2);	ERROR El propietario del departamento no puede ser huésped de la reserva

- d. Que la cantidad de huéspedes no supere la cantidad máxima de personas permitidas para una reserva

Tabla/s	Atributo/s	Chequeo	Tipo de restricción
tipo_Dpto departame nto Reserva huesped_ reserva	id_tipo_depto Id_reserva id_depto	CREATE ASSERTION AS_GR15_HUESPED_RESERVA_VALIDAR_QUE_NO_SUPERE_EL _MAX_DE_HUESPEDES CHECK NOT EXISTS (SELECT d.id_dpto, t.cant_max_huespedes,COUNT(*) FROM GR15_DEPARTAMENTO d INNER JOIN GR15_TIPO_DPTO t ON t.id_tipo_depto = d.id_tipo_depto INNER JOIN GR15_RESERVA r ON r.id_dpto = d.id_dpto INNER JOIN GR15_HUESPED_RESERVA h ON r.id_reserva = h.id_reserva GROUP BY d.id_dpto, t.cant_max_huespedes HAVING t.cant_max_huespedes<COUNT(*))	Base de datos

Prueba	Resultado esperado
INSERT INTO gr15_huesped_reserva(tipo_doc, nro_doc, id_reserva) VALUES (2, 2, 1);	ERROR: No puede haber mas huespedes de las agregados, para este tipo de departamento

4.Servicios

4.1. Por cada departamento en el sistema, de el estado en una fecha determinada, esto es si el mismo está Ocupado o Libre.

```
SELECT id_dpto,
       CASE WHEN id_dpto in (SELECT id_dpto
                             FROM GR15_RESERVA
                             WHERE fecha_desde<='2017-10-08'
                             AND fecha_hasta>='2017-10-08')
       THEN 'OCUPADO'
       ELSE 'LIBRE'
       END
       AS estado
FROM GR15_DEPARTAMENTO
```

```
CREATE OR REPLACE FUNCTION PR_GR15_obtener_estado_departamento (fecha Date)
RETURNS TABLE (
  id_dpto integer,
  estado text
)
AS $$
BEGIN
  RETURN QUERY SELECT d.id_dpto,
                     CASE WHEN d.id_dpto in (SELECT r.id_dpto
                                             FROM GR15_RESERVA r
                                             WHERE r.fecha_desde <= fecha
                                             AND r.fecha_hasta >= fecha)
                     THEN 'OCUPADO'
                     ELSE 'LIBRE'
                     END
                     END
                     AS estado
  FROM GR15_DEPARTAMENTO d;
END; $$
LANGUAGE 'plpgsql';

SELECT * FROM PR_GR15_obtener_estado_departamento ('2017-10-08')
```


4.2. Dada una rango de fechas y una ciudad, devuelva una lista de departamentos disponibles.

```
SELECT id_dpto
FROM GR15_DEPARTAMENTO
WHERE ciudad = 'Tandil'
AND id_dpto in (SELECT id_dpto
                FROM GR15_RESERVA
                WHERE fecha_desde<='2017-10-08'
                AND fecha_hasta>='2017-10-08')
```

```
CREATE OR REPLACE FUNCTION PR_GR15_obtener_departamentos_disponibles (fechaD
Date, fechaH Date, ciu varchar)
RETURNS TABLE (
id_dpto integer
)
AS $$
BEGIN
    RETURN QUERY SELECT d.id_dpto
    FROM GR15_DEPARTAMENTO d
    WHERE ciudad = ciu
    AND d.id_dpto in (SELECT r.id_dpto
                    FROM GR15_RESERVA r
                    WHERE fecha_desde <= fechaD
                    AND fecha_hasta >= fechaH);
END; $$
LANGUAGE 'plpgsql';

SELECT * FROM PR_GR15_obtener_departamentos_disponibles('2017-10-08', '2017-10-08',
'Tandil');
```

5. Definición de vistas

5.1. Devuelva un listado de todos los departamentos del sistema junto con la recaudación de los mismos en los últimos 6 meses.

```
CREATE OR REPLACE VIEW GR15_Departamentos_con_recaudacion
AS SELECT id_dpto, SUM(importe)
FROM gr15_reserva r JOIN gr15_pago p
    ON (r.id_reserva = p.id_reserva)
WHERE fecha_pago > (current_date - 180)
GROUP BY id_dpto
```

Esta vista no es actualizable ya que contiene la función de agregación SUM.

5.2. Devuelva un listado con los departamentos ordenados por ciudad y por mejor rating (estrellas)

```
CREATE OR REPLACE VIEW GR15_Departamentos_por_ciudad_y_rating
AS SELECT d.*, c.estrellas
FROM gr15_reserva r, gr15_departamento d, gr15_comentario c
WHERE r.id_dpto = d.id_dpto
    AND r.id_reserva = c.id_reserva
ORDER BY d.ciudad, c.estrellas DESC;
```

Esta vista no es actualizable ya que no conserva todas las columnas de las claves (falta la clave de la tabla comentario).

6. UI del sitio

6.1. Para un determinado departamento en un mes muestre los días que está disponible y los que está ocupado

Se creó un Modelo con la siguiente query:

```
SELECT fecha_desde, fecha_hasta FROM gr15_reserva
WHERE id_dpto=:id_dpto AND
    (EXTRACT(MONTH FROM fecha_desde)=:mes OR EXTRACT(MONTH FROM
    fecha_hasta)=:mes)
```

La primer página de la UI muestra un formulario en la que deberán ingresarse los datos de la base de datos postgres que se desea conectar. En la segunda página se debe indicar el departamento y el mes. Luego se mostrará un calendario con las fechas ocupadas y libres.

Repo: <https://github.com/samantatorrez/BasesDeDatos>