

**RF1.1** El sistema permite al administrador crear un nuevo perfil de usuario según el rol requerido junto a sus funciones y restricciones que tiene dentro del sistema

```
insert into tp_login
```

```
values (null,1122123456,'Elcacique123');
```

**RF1.2** El sistema permitirá al usuario iniciar de sesión, en donde deberá ingresar cargo al cual se desempeña, número de documento y contraseña

**RF1.3** el sistema permitirá al usuario recuperar contraseña mediante un botón de "recuperar contraseña" en llegado caso de olvidarse

**RF1.4** El sistema deberá solicitar al usuario el cambio de contraseña cada 8 semanas por seguridad de información

```
UPDATE tp_login
```

```
SET log_contraseña = 'AjiacoConCremadeLeche3154'
```

```
WHERE log_emp_numDocumento = 1029384634
```

**RF1.5** El sistema permitirá al usuario cerrar sesión mediante un botón de "Cerrar Sesión"

**RF2.1** El sistema permitirá al encargado de atención al usuario registrar un PQRS que tenga el huésped respecto a los servicios que se encuentran en el área del hotel

```
INSERT INTO tp_pqrs
```

```
VALUES (NULL, 20241018, 'El huésped reporta que la habitación no contaba con servicio de agua',20241019,1000289068, 3, 2, 2, 1);
```

**RF2.2** El sistema permitirá al de atención al usuario responder y solucionar el PQRS que tenga el huésped con respecto al área del hotel

```
INSERT INTO ti_responder
```

```
(NULL, '¡Gracias por tu comentario! Lamentamos que hayas tenido una mala experiencia en el hotel Patroclín. Tu comentario nos ayuda a mejorar nuestros servicios día a día.', 20241018, 7, 1626478765);
```

**RF3.1 El sistema permitirá al recepcionista registrar los datos personales del huésped para concretar la reserva.**

```
INSERT INTO tp_huespedes
```

```
VALUES (1000289068, 3116182673, 'Bleachowl98@gmail.com', 'Favian Alejandro', 'Machuca Pedraza', 1, 1, 4),
```

**RF3.2 El sistema permitirá al recepcionista generar una factura para el huésped en la que se señale que la reserva ha sido concretada de manera exitosa.**

```
INSERT INTO tp_reservas
```

```
VALUES (NULL, 350000.00, 20240624, 20240626, 1, 1, 10, 1029384634, 1, 1002455665),
```

**RF3.3 El sistema permitirá al huésped recibir un correo con su reserva al huésped señalando que la reserva ha sido concretada de manera exitosa.**

```
SELECT r.res_id AS id,  
r.res_costo AS Costo,  
r.res_fechainicio AS Inicio_reserva,  
r.res_fechaFin AS Fin_reserva,  
r.res_canPersonas AS Cantidad_personas,  
m.mot_descripcion AS Motivo_reserva,  
hab.hab_numero AS Número_habitación,  
t.tiphab_descripcion AS Tipo_habitación,  
es.estres_descripcion AS Estado_reserva,  
h.hue_nombres AS Nombres_huesped,
```

```

h.hue_apellidos AS Apellidos_huesped,
e.emp_nombres AS Nombres_empleado,
e.emp_apellidos AS Apellidos_empleado
FROM tp_reservas r
INNER JOIN td_motivoreserva m ON r.res_mot_motivoReserva = m.mot_id
INNER JOIN tp_empleados e ON r.res_emp_numdocumento = e.emp_numDocumento
INNER JOIN td_estadoreserva es ON r.res_estres_estado = es.estres_id
INNER JOIN tp_huespedes h ON r.res_hue_numdocumento = h.hue_numDocumento
INNER JOIN tp_habitaciones hab ON hab.hab_numero = r.res_hab_numero
INNER JOIN td_tipohabitacion t ON hab.hab_tipohab_tipoHabitacion = t.tiphab_id

```

#### **RF3.4 El sistema permitirá al recepcionista hacer cambios en la reserva**

```

UPDATE tp_reservas

SET res_fechainicio = 20240630, res_fechaFin = 20240703

WHERE res_id = 13

```

#### **RF3.5 El sistema permitirá al huésped recibir por correo que su servicio ha sido modificado**

```

SELECT r.res_id AS id,
r.res_costo AS Costo,
r.res_fechainicio AS Inicio_reserva,
r.res_fechaFin AS Fin_reserva,
r.res_canPersonas AS Cantidad_personas,
m.mot_descripcion AS Motivo_reserva,
hab.hab_numero AS Número_habitación,
t.tiphab_descripcion AS Tipo_habitación,
es.estres_descripcion AS Estado_reserva,
h.hue_nombres AS Nombres_huesped,
h.hue_apellidos AS Apellidos_huesped,
e.emp_nombres AS Nombres_empleado,

```

```

e.emp_apellidos AS Apellidos_empleado
FROM tp_reservas r
INNER JOIN td_motivoreserva m ON r.res_mot_motivoReserva = m.mot_id
INNER JOIN tp_empleados e ON r.res_emp_numdocumento = e.emp_numDocumento
INNER JOIN td_estadoreserva es ON r.res_estres_estado = es.estres_id
INNER JOIN tp_huespedes h ON r.res_hue_numdocumento = h.hue_numDocumento
INNER JOIN tp_habitaciones hab ON hab.hab_numero = r.res_hab_numero
INNER JOIN td_tipohabitacion t ON hab.hab_tiphab_tipoHabitacion = t.tiphab_id
WHERE res_id = 13

```

**RF3.6 El sistema le permitirá al recepcionista la cancelación de su reserva**

```

UPDATE tp_reservas

SET res_estres_estado = 3

WHERE res_id = 13

```

**RF 3.7 El sistema permitirá al huésped recibir una notificación al correo que su servicio ha sido cancelado.**

**RF 4.1 El sistema permitirá al recepcionista observar la información detallada sobre una habitación**

```

SELECT h.hab_numero as Numero_Habitacion, h.hab_costo as Costo, h.hab_capacidad as
Capacidad_Personas, t.tiphab_descripcion as Tipo_Habitacion, ñ.tam_descripcion as
Tamaño_Habitacion, e.esthab_descripcion as Estado_Habitacion
FROM tp_habitaciones h
INNER JOIN td_tipohabitacion t ON h.hab_tiphab_tipoHabitacion = t.tiphab_id
INNER JOIN td_tamaño ñ ON h.hab_tam_tamaño = ñ.tam_id
INNER JOIN td_estadohabitacion e ON h.hab_esthab_estado = e.esthab_id;

```

**RF 4.2El sistema permitirá al recepcionista apartar una habitación.**

```

UPDATE tp_habitaciones

SET hab_esthab_Estado = 1

```

WHERE hab\_numero = 69

**RF 4.3 El sistema debe permitir al recepcionista filtrar las habitaciones**

```
SELECT h.hab_numero AS Habitación,  
h.hab_costo AS Costo,  
h.hab_capacidad AS Capacidad,  
th.tiphab_descripcion AS Tipo_Habitación,  
t.tam_descripcion AS Tamaño,  
eh.esthab_descripcion AS Estado_habitación  
FROM tp_habitaciones h  
INNER JOIN td_tipohabitacion th ON h.hab_tiphab_tipoHabitacion = th.tiphab_id  
INNER JOIN td_tamaño t ON h.hab_tam_tamaño = t.tam_id  
INNER JOIN td_estadohabitacion eh ON h.hab_esthab_estado = eh.esthab_id  
WHERE h.hab_esthab_estado = 2
```

**RF 4.4 El sistema debe permitir al recepcionista cambiar el estado de la habitación**

```
UPDATE tp_habitaciones  
SET hab_esthab_Estado = 1  
WHERE hab_numero = 69
```

**RF 4.5 El sistema debe permitir al recepcionista modificar la habitación.**

```
UPDATE tp_habitaciones h  
SET h.hab_costo = 350000  
WHERE h.hab_numero = 10
```

**RF 5.1 El sistema debe permitir al administrador consultar la información de todos los perfiles de los usuarios**

```

SELECT

l.log_id AS Id_cuenta,

e.emp_numDocumento AS Número_documento,

e.emp_nombres AS Nombres,

e.emp_apellidos AS Apellidos,

r.rol_descripcion AS Rol,

l.log_contraseña AS Contraseña

FROM tp_empleados e

INNER JOIN tp_login l ON l.log_emp_numDocumento = e.emp_numDocumento

INNER JOIN td_rols r ON e.emp_rol_rols = r.rol_id;

```

**RF 5.2 El sistema permitirá al administrador editar la información de los usuarios, como su rol y funciones dentro del sistema**

```

UPDATE tp_empleados e

SET e.emp_rol_rols = 4

WHERE e.emp_numDocumento = 1029384634

```

**RF 5.3 el sistema permitirá al administrador bloquear o deshabilitar el perfil de un usuario, impidiendo así al usuario ingresar al sistema**

**RF 5.4 El sistema permitirá al administrador eliminar el perfil de un usuario, dejándola en estado de reposo durante un tiempo de 30 días hasta su eliminación completa, con la posibilidad de deshacer el cambio.**