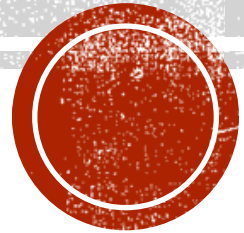
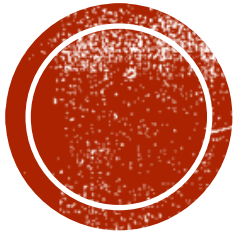
TAQUILLA VIRTUAL

Diseño detallado y plan de pruebas





DISEÑO DETALLADO

ESPECTACULO

id_espectaculo (int)
descripcion (varchar)
nombre (varchar)
tipo (varchar)

PARTICIPANTE

id_participante (int)
nombre(varchar)

PARTICIPANTE-ESPECTACULO

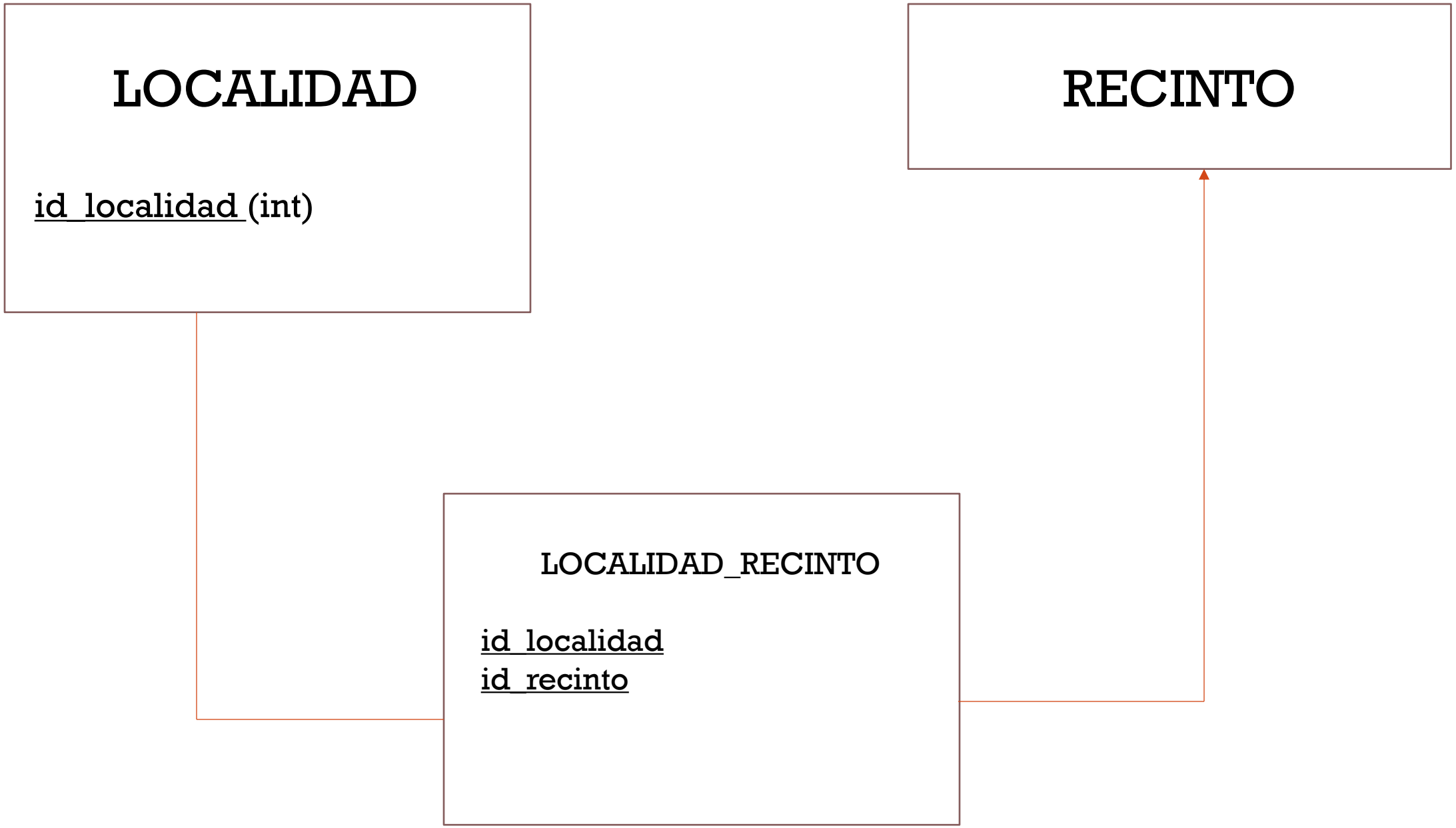
id_espectaculo,id_participante

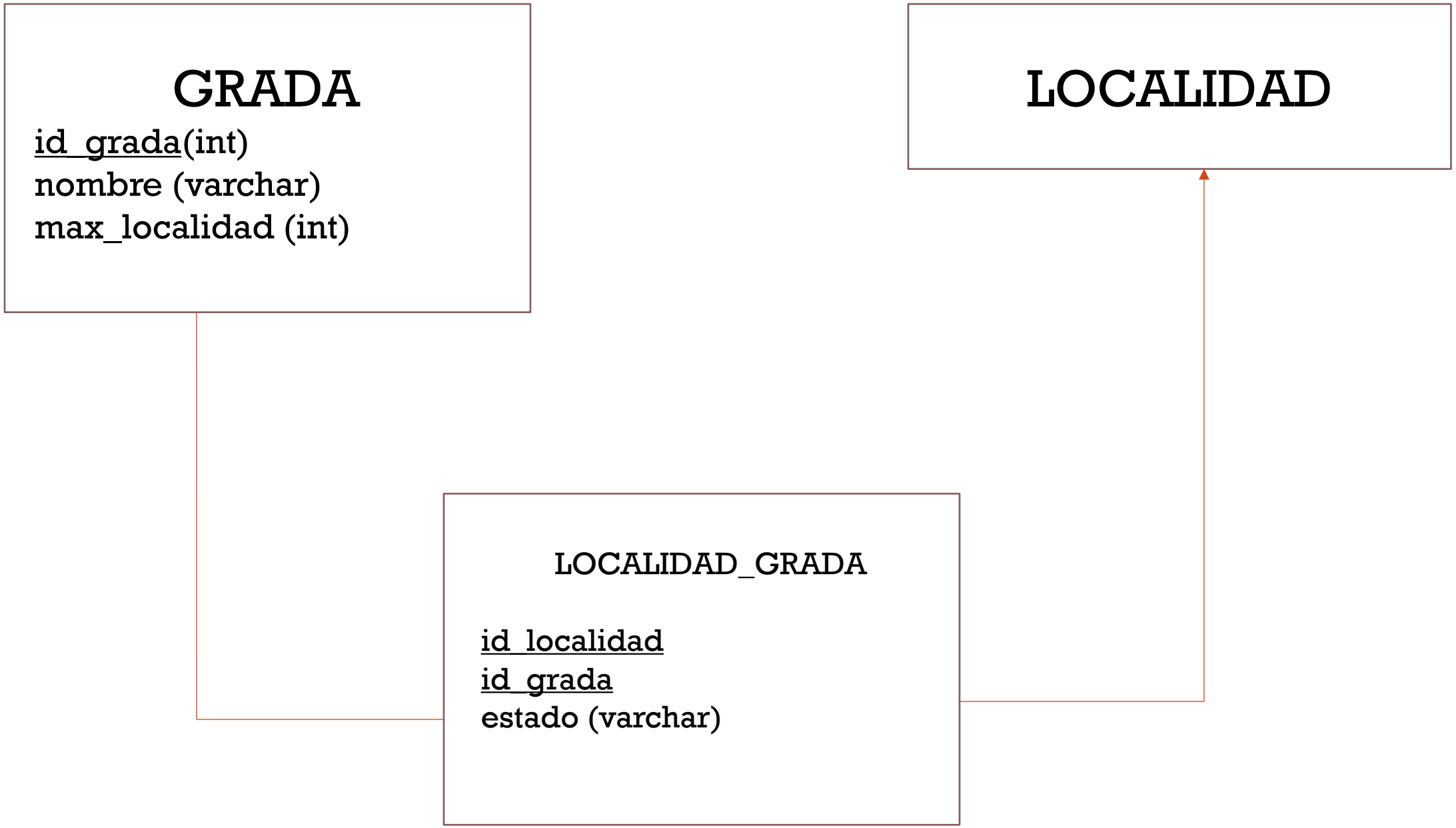
RECINTO

id_recinto (int)
nombre (varchar)

ESPECTACULO

EVENTO
id_evento (int)
id_espectaculo (int)
id_recinto (int)
fecha (Date)
estado (varchar)


LOCALIDAD
id_localidad (int)
RECINTO
LOCALIDAD_RECINTO
id_localidad
id_recinto


GRADA
id_grada(int)
nombre (varchar)
max_localidad (int)
LOCALIDAD
LOCALIDAD_GRADA
id_localidad
id_grada
estado (varchar)

GRADA

EVENTO
(agregación)

GRADA_EVENTO

id_espectaculo
id_recinto
id_evento
id_grada

LOCALIDAD-GRADA (agregación)

EVENTO (agregación)

PRECIO

id_evento
id_recinto
id_espectaculo
id_grada
id_localidad
tipo_usuario

USUARIO

tipo (varchar)

LOCALIDAD-GRADA

CLIENTE

id_cliente
datos_personales (varchar)*
datos_bancarios (varchar)*

RESERVA

id_cliente
id_localidad
id_grada
tipo (varchar)

NOTA: datos personales y bancarios están simplificados, serían más complejos.

PROCEDIMIENTOS ALMACENADOS

- ver_eventos(IN limite (int), IN pagina (varchar))
  - Devuelve N eventos saltándose N*M. (N = limite, M=pagina)
- buscar_evento(IN nombre(varchar))
  - Devuelve los eventos que empiecen por el nombre introducido.
- ver_gradas(IN id_evento(int))
  - Devuelve las gradas disponibles para el evento solicitado.
- ver_localidades(IN id_grada(int), IN tipo (varchar), OUT precio (decimal))
  - Devuelve las localidades correspondientes a la grada solicitada y su precio.
- crear_cliente(IN dni(int), IN datos_personales (varchar), IN datos_bancarios(varchar))
  - Introduce un nuevo cliente en la base de datos y no devuelve nada.

PROCEDIMIENTOS ALMACENADOS

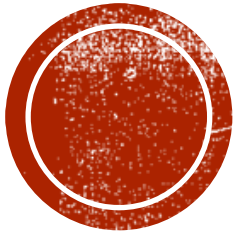
- consultar_usuario(IN dni(int))
  - Devuelve los datos personales del usuario solicitado.
- prerreservar_localidad(IN id_grada (int), IN id_localidad(int), IN dni(int))
  - Se realiza la prerreserva de una localidad para una grada y evento determinado, no devuelve nada.
- reservar_localidad(IN id_grada (int), IN id_localidad(int), IN dni(int))
  - Se realiza la reserva de una localidad para una grada y evento determinado, no devuelve nada.
- ver_localidades_prerreservadas(IN dni(int))
  - Devuelve las localidades prerreservadas por el usuario indicado.
- ver_localidades_reservadas(IN dni(int))
  - Devuelve las localidades reservadas por el usuario indicado.
- anular_reserva(IN id_localidad(int),IN id_grada(int), IN dni(int))
  - Anula una reserva o una pre-reserva de una localidad.

TRIGGERS

- Al introducir una reserva/pre-reserva se dispara un trigger para cambiar el estado de la localidad sobre la que se realiza la reserva/pre-reserva y, en caso de que todas las localidades estén reservadas/pre-reservadas, el estado del evento pasará a cerrado.
- Al anular una reserva/pre-reserva se dispara un trigger para cambiar el estado de la localidad sobre la que se realiza la reserva/pre-reserva y, en caso de que el evento estuviese completo, pasará a estar no abierto.

EVENTOS

- El trigger que se dispara al introducir una reserva/pre-reserva programa que se lance un evento t1 minutos después de la introducción. En el momento en el cual salte el evento, esa reserva/pre-reserva se elimina.



PLAN DE PRUEBAS

PRUEBAS

1. Hay 10 eventos en la base de datos y se solicita la página 1 con límite 3.
    1. Deberían aparecer los eventos 3,4 y 5.
    2. En caso de que no haya suficientes eventos devuelve un set vacío.
2. Hay X eventos en la base de datos y hay 2 eventos cuyo nombre empieza por "concierto". Se hace una búsqueda por "concierto".
    1. Deberían aparecer los dos eventos cuyo nombre empieza por concierto.
    2. En caso de que no hubiera eventos cuyo nombre empieza por la búsqueda se devuelve un set vacío.
3. Se realiza una búsqueda de las gradas del evento con id=X.
    1. Si existe el evento, se devolverá el set con las gradas correspondientes.
    2. En caso contrario, se devolverá un error.

PRUEBAS

4. Se realiza la búsqueda de las localidades de una grada con id=X y para el tipo=Y.
    1. Si existe tanto la grada como el tipo de usuario para esa grada, se devuelven el set de localidades y el precio asociado.
    2. Si no existe la grada o el tipo seleccionado, se devuelve un error.
5. Se crea un cliente con una serie de datos y un DNI=X.
    1. Si el cliente no existe, se crea uno nuevo.
    2. Si el cliente ya existe, se devuelve un error.
6. Se consultan los datos de un cliente con DNI=X.
    1. Si el cliente existe, se devuelven los datos de ese cliente.
    2. Si el cliente no existe, se devuelve un error.

PRUEBAS

7. Se realiza una pre-reserva para una grada con id=X, para una localidad=Y y para un DNI=Z y para un tipo de usuario=A.
    1. Si la grada, la localidad y el cliente existen, la localidad está disponible para ser pre-reservada y esa grada está disponible para ese tipo de usuario, el estado de la localidad cambia a pre-reservado y a ese cliente se le asocia una pre-reserva.
    2. Si faltan menos de t2 minutos para que comience el evento, no se permite la pre-reserva.
    3. En otro caso, se devuelve un error.
8. Se realiza una reserva para una grada con id=X, para una localidad=Y y para un DNI=Z y para un tipo de usuario=A.
    1. Si la grada, la localidad y el cliente existen, la localidad está disponible para ser pre-reservada y esa grada está disponible para ese tipo de usuario, el estado de la localidad cambia a reservado y a ese cliente se le asocia una reserva.
    2. Si el evento ya ha comenzado, no se permite realizar la reserva.
    3. En otro caso, se devuelve un error.

PRUEBAS

9. Se realiza la anulación de una pre-reserva para una localidad con id_localidad=X, id_grada=Y y cliente con DNI=Z.
    1. Si existen la localidad, la grada y el cliente, y ese cliente había pre-reservado la localidad, se realizará la anulación y el estado de la localidad cambia a libre.
    2. En caso contrario, se devuelve un error.
10. Se realiza la anulación de una pre-reserva automáticamente al sobrepasarse el tiempo de t1 minutos desde que se realizó la pre-reserva.
11. Se realiza la anulación de una reserva ara una localidad con id_localidad=X, id_grada=Y y cliente con DNI=Z.
    1. Si existen la localidad, la grada y el cliente, y ese cliente había reservado la localidad, se realizará la anulación y el estado de la localidad cambia a libre.
    2. En caso contrario, se devuelve un error.

PRUEBAS

12. Se solicitan las localidades pre-reservadas por un cliente con DNI=X.
    1. Si existe el cliente y además tiene localidades pre-reservadas, se devuelven esas localidades.
    2. En caso contrario, se devuelve un error.
13. Se solicitan las localidades reservadas por un cliente con DNI=X.
    1. Si existe el cliente y además tiene localidades reservadas, se devuelven esas localidades.
    2. En caso contrario, se devuelve un error.

- María Losada González
- Brais Piñeiro Fraga
- Pablo Rodríguez Pérez
- Pablo Sio Fernández