

# Seminario de Lenguajes

## Opción: PHP, React y API Rest

### 2024

#### API REST / Slim

Slim es un micro framework de PHP que se utiliza para crear aplicaciones web y APIs REST.

Las API REST siguen el modelo CRUD, que significa Create (Crear), Retrieve (Recuperar), Update (Actualizar) y Delete (Eliminar)

Los recursos de una API REST se identifican mediante endpoints o URI (Uniform Resource Identifier). Cada recurso tiene una dirección única a la que se puede acceder para realizar operaciones sobre él.

Generalmente, se utiliza JSON (JavaScript Object Notation) como formato para el intercambio de datos entre el cliente y el servidor.

#### ACLARACIONES

Se deben tener las siguientes consideraciones:

- Todos los endpoints deberán tener los métodos y nombres que se detallan en cada ítem.  
Por ejemplo **POST** /localidad/crear -> **Método** y **Nombre** del endpoint
- Cuando en el endpoint que se pide en el TP se necesite valores variables que serán enviados por la URL se definirán con el signo \$  
Por ejemplo Editar: /localidad/\$id/editar -> \$id es una variable cuyo nombre es id
- Las eliminaciones sólo se podrán ejecutar correctamente en los casos en que el dato no esté siendo utilizado en otra tabla, caso contrario se debe informar el error.

## IMPORTANTE

En todos los casos, los endpoints deberán devolver el estado de la petición realizada

- 200 OK: Indica que la solicitud se procesó correctamente y se devolvió una respuesta exitosa.
- 400 Bad Request: Se utiliza cuando la solicitud enviada por el cliente es incorrecta o no se puede procesar debido a parámetros faltantes, valores inválidos o problemas de formato.
- 404 Not Found: Indica que el recurso solicitado no se pudo encontrar en el servidor. No existe el endpoint o el recurso (Por ejemplo, se buscó una localidad con un id que no existe)

En los casos en que se produce un error, además, deberán devolver un mensaje indicando por qué no se puede realizar la acción.

Por ejemplo:

- No se puede eliminar una localidad porque se está utilizando en otro registro
- No se puede editar una reserva porque ya está en curso

## PROYECTO

Se desea desarrollar una aplicación web que permita gestionar alquileres temporarios de una inmobiliaria.

Un empleado debe poder administrar las propiedades de la inmobiliaria y llevar el control de las reservas, así como también de las personas que las realizan (inquilinos).

Los nombres de las localidades y de los tipos de propiedades no se pueden repetir.

El documento de un inquilino no se puede repetir.

En cada reserva, el valor total es la multiplicación del valor de una propiedad por una noche por la cantidad de noches de la reserva.

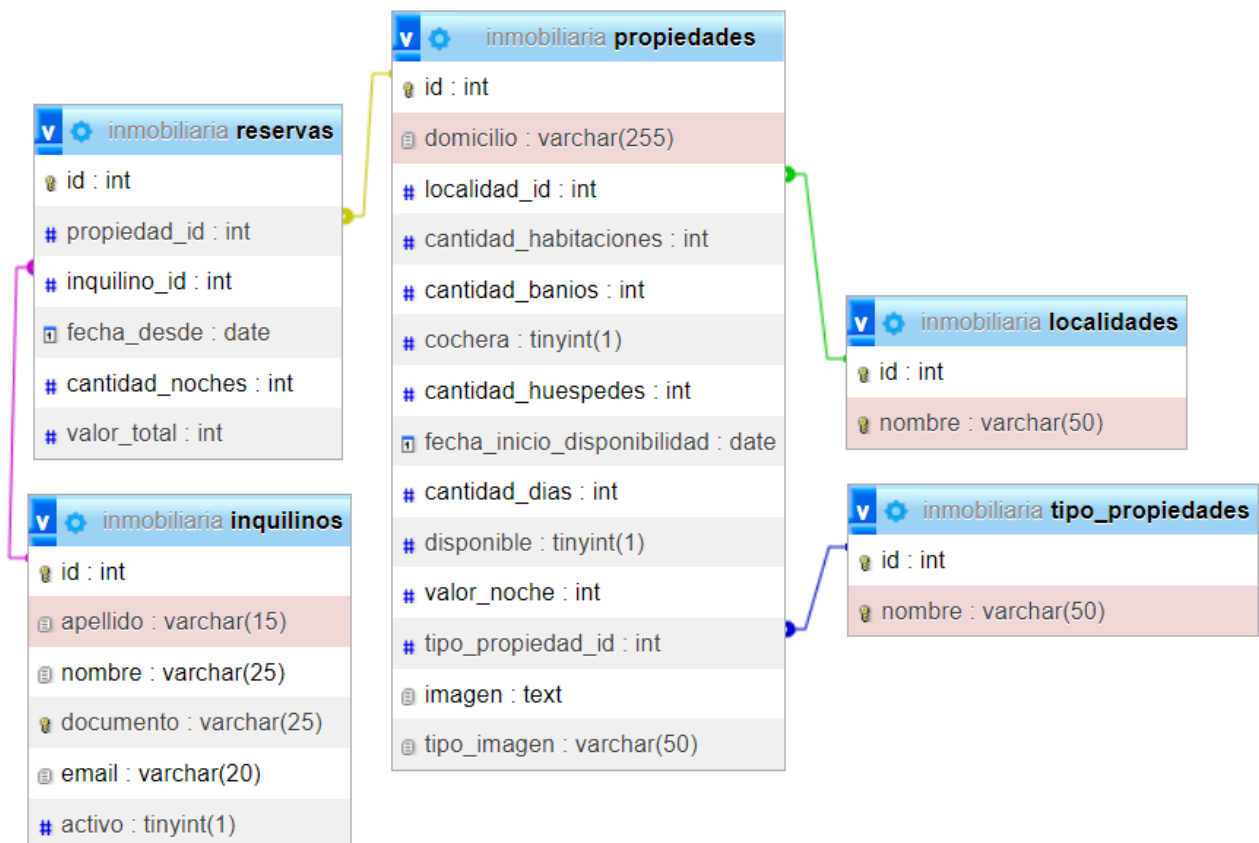
Una reserva solo se puede realizar si el inquilino está activo y la propiedad está disponible.

Una reserva solo se puede editar o eliminar si no comenzó (fecha\_desde es menor a la fecha actual)

Ejemplo de una propiedad que se puede editar y/o eliminar:

fecha\_desde: 10/2/2024

fecha actual: 8/2/2024



## PARTE UNO

Para la aplicación, es necesario desarrollar los siguientes endpoints:

### 1. Localidades:

El nombre debe ser único

Endpoints

- a. Crear: POST /localidades
- b. Editar: PUT /localidades/\$id
- c. Eliminar: DELETE /localidades/\$id
- d. Listar: GET /localidades

Campos:

id: integer  
nombre: string (50) requerido

### 2. Tipo Propiedad:

El nombre debe ser único

Endpoints

- a. Crear: POST /tipos\_propiedad
- b. Editar: PUT /tipos\_propiedad/\$id
- c. Eliminar: DELETE /tipos\_propiedad/\$id
- d. Listar: GET /tipos\_propiedad

Campos:

id: integer  
nombre: string (50) requerido

### 3. Inquilino:

El documento debe ser único.

Endpoints

- a. Crear: POST /inquilinos
- b. Editar: PUT /inquilinos/\$id
- c. Eliminar: DELETE /inquilinos/\$id
- d. Listar: GET /inquilinos
- e. Ver inquilino: GET /inquilinos/\$id
- f. Historial: muestra el historial de reservas de un inquilino  
GET /inquilinos/\$idInquilino/reservas

Campos: **Todos los datos son requeridos**

id: integer  
nombre\_usuario: string (20)  
apellido: string (15)  
nombre: string (25)  
email: string (20)  
activo: boolean

#### 4. Propiedad

Se debe poder filtrar el listado de propiedades por los siguientes criterios (se indican con **rojo** los nombres que se deben respetar)

- Disponible (true/false) -> **disponible**
- Localidad: id (int) -> **localidad\_id**
- Fecha de inicio: date -> **fecha\_inicio\_disponibilidad**
- Cantidad de huéspedes: int -> **cantidad\_huespedes**

#### Endpoints

- Crear: POST /propiedades
- Editar: PUT /propiedades/\$id
- Eliminar: DELETE /propiedades/\$id
- Listar: GET /propiedades
- Ver propiedad: GET /propiedades/\$id

#### Campos

id: integer

domicilio: string (**requerido**)

localidad\_id: integer (se elige de la tabla de localidades, **requerido**)

cantidad\_habitaciones: integer

cantidad\_banios: integer,

cochera: booleano (true/false)

cantidad\_huespedes: integer (**requerido**),

fecha\_inicio\_disponibilidad: date (**requerido**)

cantidad\_días: integer (**requerido**)

disponible: boolean (**requerido**)

valor\_noche: integer (**requerido**)

moneda\_id: integer (se elige de la tabla de monedas, **requerido**)

tipo\_propiedad\_id: (se elige de la tabla de tipo\_propiedad, **requerido**)

imagen: text

tipo\_imagen: string

#### 5. Reserva:

El valor total es la multiplicación del valor de una propiedad por una noche por la cantidad de noches de la reserva

Una reserva solo se puede realizar si el inquilino está activo y la propiedad está disponible.

Una reserva solo se puede editar o eliminar si no comenzó (fecha\_desde es menor a la fecha actual)

Ejemplo de una propiedad que se puede editar y/o eliminar:

fecha\_desde: 10/2/2024

fecha actual: 8/2/2024

#### Endpoints:

- Crear: POST /reservas
- Editar: PUT /reservas/\$id
- Eliminar: DELETE /reservas/\$id

d. Listar: GET /reservas

Campos: **Todos los datos son requeridos.**

Id: integer

propiedad\_id: integer (se elige de la tabla de propiedad)

inquilino\_id: (se elige de la tabla de inquilino)

fecha\_desde: date

cantidad\_noches: integer

valor\_total: integer.

## METODOLOGÍA PARA LA ENTREGA EN IDEAS

- Hacer un archivo zip o rar con todos los archivos del proyecto
- Subir a la sección del grupo asignada (En el repositorio general NO)
- **Fecha de entrega: 30/4**