

# Se debe construir un api para pagos de arrendamientos

## Restricciones técnicas:

1. El api debe correr en el puerto 8084.
2. Debe respetar **exactamente** la firma enviada (los paths de los endpoints, las estructura de entradas/salidas y los nombre de los campos de los jsons). **Nota: Únicamente se debe crear un post y un get respetando la firma solicitada.**
3. Se debe usar la base de datos mysql y usar desde la aplicación el usuario "ceiba" con la clave "ceiba". (es decir en el connection string de la base de datos).
4. La información **se debe almacenar en la base de datos mysql** creando un schema llamado "pruebaceiba".
5. Únicamente se debe crear una tabla llamada "pagos" los 4 campos de la tabla se deberán llamar igual a los campos del objeto json (más adelante la información).
6. Se debe de enviar un archivo **README** en el cual se especifique como levantar el api por ejemplo "npm start".
7. Usar para resolver el problema a nivel de arquitectura una de estas opciones obligatoriamente: **mvc, clean architecture o arquitectura hexagonal.**
8. Construir pruebas unitarias automatizadas de la lógica de negocio con alguna de las siguientes herramientas dependiendo del lenguaje: junit(java), jest (node), golang testing (golang) o unittest (phyton).
9. Si usted trabaja en node se debe construir usando el framework **nest js o con la librería express con typescript**, si usted trabaja en golang se debe construir usando el framework **gin gonic**, si usted trabaja en phyton se debe construir usando el framework **flask**, si usted

trabaja en java se debe construir usando el framework **spring boot**.

10. La solución del problema se debe enviar en un archivo .zip y **no debe publicarse en repositorios de código público** como github, gitlab etc (Se penalizará esto).

### **Reglas de negocio**

1. El formato de la "fechaPago" debe ser dd/mm/yyyy, es decir no se permite enviar fechas en formatos diferentes. **En este caso se debe retornar el código http 400 (bad request) con el mensaje "Formato de fecha incorrecto"**
2. El campo "documentoIdentificacionArrendatario" debe ser solo numérico.
3. El campo "codigoInmueble" debe ser alfanumérico.
4. El campo "fechaPago" debe ser siempre una fecha válida y existente en el calendario.
5. El campo "valorPagado" deberá estar entre 1 y 1000000 (1 millón) es decir se reciben pagos parciales.
6. El valor del arriendo total mensual para todos los arrendatarios es de 1000000 (1 millón).
7. Si el pago se hace total es decir 1000000 (1 millón) la respuesta de la operación deberá ser "gracias por pagar todo tu arriendo".
8. Si el pago se hace parcial es decir menos de 1000000 (1 millón) la respuesta de la operación deberá ser "gracias por tu abono, sin embargo recuerda que te hace falta pagar \${poner acá el valor restante}".
9. Si el pago se hace por el valor restante la respuesta de la operación deberá ser "gracias por pagar todo tu arriendo".
10. Solo se reciben pagos los días impares de cada mes es decir si se envía en el campo "fechaPago" el valor "02/09/2020" la respuesta debe ser "lo siento pero no se

puede recibir el pago por decreto de administración"; **en este caso se debe retornar el código http 400 (bad request)**

11. El identificador único de cada arrendatario es "documentoIdentificacionArrendatario" y "codigoInmueble"

**Importante:** Ubique las reglas de negocio en la capa de dominio si está usando arquitectura hexagonal o arquitectura limpia o si está usando mvc en la capa del modelo, evite poner lógica en los controladores o persistencia.

## ejemplos de firmas

**POST /api/pagos (permitirá el registro de pagos)**

json de entrada

```
{
  "documentoIdentificacionArrendatario": "1036946622",
  "codigoInmueble": "8870",
  "valorPagado": "1000000",
  "fechaPago": "25/09/2020"
}
```

json de salida

```
{
  "respuesta": "gracias por pagar todo tu arriendo"
}
```

**código http resultante: 200**

**GET /api/pagos (permitirá consultar el listado de pagos realizados)**

ejemplo aleatorio json de salida

```
[
  {
    "documentoIdentificacionArrendatario": "1036946622",
    "codigoInmueble": "8870",
    "valorPagado": "1000000",
  }
]
```

```
    "fechaPago": "25/09/2020"
  },
  {
    "documentoIdentificacionArrendatario": "1036946623",
    "codigoInmueble": "8871",
    "valorPagado": "500000",
    "fechaPago": "25/09/2020"
  }
]
```

**código http resultante: 200**