



**Certified Tech  
Developer**  
The Ultimate Degree

# Programación Imperativa

## Práctica guiada

### Objetivo

Integrar los conocimientos adquiridos hasta el momento en la cursada.

Practicar la lectura, comprensión e interpretación de consignas y producir código que resuelva los enunciados propuestos.

### Buenas prácticas

A la hora de la resolución no solo es importante que el código funcione correctamente, sino también que sea legible y respete las buenas formas.

- Que las variables, métodos y funciones tengan nombres descriptivos
- Que se utilicen nombres en español o en inglés pero no ambos
- Que se utilice camelCase donde corresponda
- Que se utilicen los métodos más adecuados para resolver cada problema
- Que el código produzca el resultado esperado a partir de los datos suministrados

## Introducción

Una empresa inmobiliaria dedicada a la renta / alquiler de departamentos necesita crear un pequeño sistema que le permita facilitar parte de su operatoria diaria.

El listado de los departamentos lo obtendremos de un archivo con formato JSON que contendrá un array de objetos literales que representarán los departamentos.

De cada departamento podremos saber:

- Él id de departamento, será un identificador.
- La cantidad de habitaciones
- La cantidad máxima de personas
- Si está o no disponible para el alquiler,
- Si se aceptan o no las mascotas,
- El valor del alquiler
- Los comentarios que tiene el departamento



## Consignas

### 1. Obtener el listado de los departamentos.

Tomando como base el [siguiente archivo JSON](#), deberán:

- Descargar el archivo y colocarlo en la raíz del proyecto
- En un nuevo archivo app.js, leer los contenidos del archivo y parsearlos
- En base a lo anterior, guardar el resultado en una variable

**Resultado esperado:** variable conteniendo un array con todos los departamentos. Recuerden que pueden verificar el contenido de la variable haciendo un console.log().

### 2. Crear un objeto literal que represente la inmobiliaria

Este objeto contendrá, una propiedad con el listado de departamentos obtenido en el punto anterior y las **funcionalidades** que nos solicitan a continuación:

- Agregar una propiedad llamada **departamentos** que contenga los departamentos.
- Agregar un método **departamentosDisponibles** que permita consultar disponibilidad, es decir, que devuelva una lista de los departamentos disponibles.
- Agregar un método **listarDepartamentos** que reciba como parámetro un array de departamentos y los imprima por consola.

Este método deberá imprimir por cada elemento:

- El id.



- El precioAlquiler.
- disponible (en caso de ser true mostrar "Disponible" caso contrario "Alquilado")

ej: "id: 2, precio \$ 7500, está Disponible"

- D. Agregar un método **buscarPorId** que permita buscar el departamento en función de su identificador.
- Este método debe devolver un departamento en caso de encontrar el identificador.
- E. Agregar un método **buscarPorPrecio** que permita filtrar los departamentos disponibles, siempre y cuando el precio sea menor o igual al precio enviado como argumento.
- Este método debe usar **departamentosDisponibles**.
  - Este método debe devolver una lista de departamentos.
- F. Agregar un método que permita **ordenar por precio** de menor a mayor según el precioAlquiler y devolverlo.
- G. Agregar un método **precioConImpuesto** que permita realizar un incremento del precioAlquiler a todos los departamentos, *tanto disponibles como no disponibles*, según un porcentaje que se envía como argumento al momento de la invocación del método.
- ejemplo : PrecioConImpuesto(10), deberá incrementar el precioAlquiler de Todos los departamentos un 10%

**Resultado esperado:** un array con los precios con el impuesto sumado;



H. Agregar un método que permita **alquilar** un departamento.

- Este método recibe por parámetro el id del departamento y actualiza la propiedad del departamento en false.

**Resultado esperado:** que quede actualizada la propiedad departamentos de la inmobiliaria, donde el departamento que alquilo no está disponible.

I. Agregar un método que permita **calcular el ingreso** en el flujo de caja que tendría la empresa cuando se pague el alquiler de los departamentos.

**Resultado esperado:** suma del alquiler de todos los departamentos que estén alquilados.



### 3. Métodos adicionales

A. Agregar un método **comentar** que permita realizar un comentario, (*la propiedad comentarios es un array de objetos dentro de el objeto departamento*) a un departamento indicando los siguientes datos:

- comentario = string
- puntuacion = number (1 ... 5) .

**Resultado esperado:** un mensaje "Comentario realizado con éxito";

B. Agregar un método **preferenciasCliente** que permita filtrar los departamentos disponibles, si cumplen con las especificaciones deseadas por el cliente. Este método deberá tomar un array de departamentos disponibles retornados del método **departamentosDisponibles**, luego realizar una iteración del mismo y retornar cuando encuentre coincidencias con las especificaciones deseadas por él cliente

especificaciones:

- cantidadHabitacion = number (*si es 0 es un monoambiente*)
- mascotas = booleano
- cantidadPersonas = number .
- precioAlquiler = number.

el método debe recibir las 4 especificaciones del cliente, y debe encontrar los departamentos que cumplan con todas ellas.

**Resultado esperado:** array de departamentos disponibles que además dispongan de las características preferidas por el cliente.