

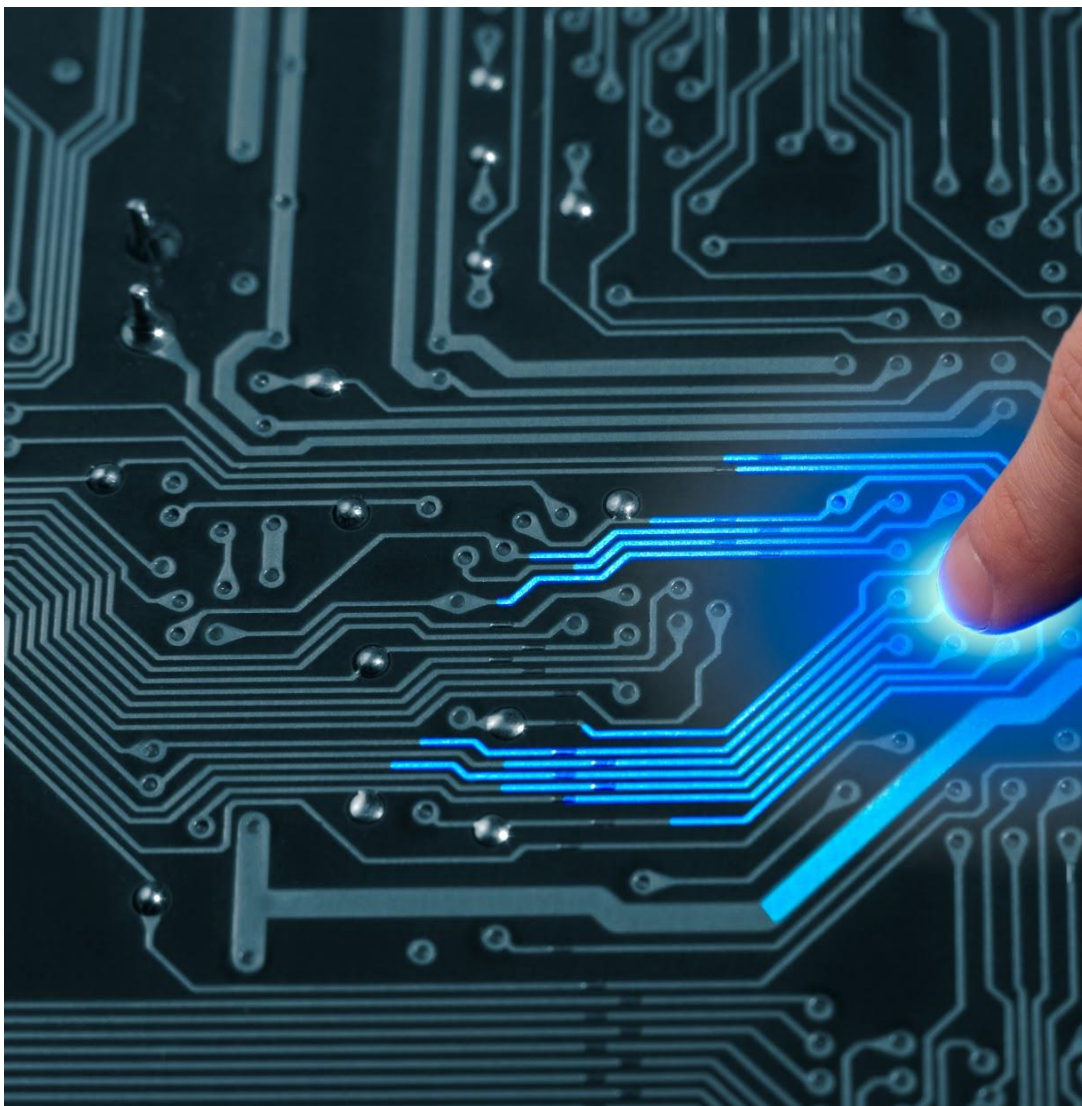
CFGS

# DESARROLLO DE APLICACIONES MULTIPLATAFORMA

PROGRAMACIÓN B

PROGRAMACIÓN ORIENTADA A OBJETOS. LIBRERÍAS DE CLASES  
FUNDAMENTALES

PAC 3: Estructuras de almacenamiento



## PAC 3: Estructuras de almacenamiento

### INFORMACIÓN

Para responder a las siguientes cuestiones deberás ayudarte del material didáctico y consultar internet.

Requisitos varios que deben cumplirse en vuestros trabajos:

- En los ejercicios, si se requieren de cálculos, estos deben aparecer en la respuesta que planteéis.
- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado aquella información.
- Siempre que utilicéis información del material didáctico para responder / resolver alguna pregunta, tenéis que citar el tema y la página de dónde habéis sacado aquella información.
- No se aceptarán respuestas sacadas de Internet utilizando la metodología de copiar y pegar. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas ha de ser vuestro.
- Las respuestas a las preguntas deben estar bien argumentadas, no se admiten respuestas escuetas o monosílabas.
- Si el proyecto no compila, la puntuación máxima será un 4.
- Se valorará la presentación, ortografía y gramática de vuestro trabajo hasta con un punto de la nota final.

Entrega de la PAC:

- Se debe entregar junto con el proyecto comprimido, una memoria en el que se expliquen los pasos realizados junto con los puntos más importantes del código explicados.

### 1. Vamos a trabajar con los datos de la clínica Vetllerna.

- En el cliente, se añadirá un ArrayList de la clase Mascota. En los dos constructores, se inicializará el ArrayList, pero no se introducirán datos, ni se recibirá ningún ArrayList por parámetro.
- Se creará un método para añadir mascotas al ArrayList.
- En la mascota, se añadirá un HashMap para almacenar los tratamientos, siendo ambos tipos (fecha y tratamiento) de tipo String. En los dos constructores, se inicializará el HashMap, pero no se introducirán datos, ni se recibirá ningún HashMap por parámetro.
- Se creará un método para añadir mascotas al HashMap.

En registro, cuando el usuario pulse en el botón de Insertar:

- Se creará un objeto de la clase Mascota con los datos de las casillas correspondientes.
- Se creará un objeto de la clase Cliente con los datos de cliente (nombre y DNI, dirección no es suficiente.)
- Se añadirá la mascota al cliente.

- Se mostrará esta información por consola.

En AñadirMascota, cuando el usuario pulse en el botón de Buscar:

- Se buscará el DNI en el ArrayList de clientes. Si lo encuentra, pondrá el nombre correspondiente en el JTextField correspondiente.

En AñadirMascota, cuando el usuario pulse en el botón de Insertar:

- Se añadirá la mascota al ArrayList de mascota del cliente con el que coincida el DNI. Si no existe el DNI pondrá un mensaje por consola.
- Se mostrará esta información por consola.

En Tratamientos, cuando el usuario pulse en el botón de Buscar:

- Se buscará el DNI en el ArrayList de clientes. Si lo encuentra, pondrá los nombres de las mascotas en el JComboBox.

En Tratamientos, cuando el usuario pulse en el botón de Añadir:

- Se añadirá el tratamiento en el HashMap de tratamientos de la mascota seleccionada en el JComboBox.
- Se mostrará esta información por consola.

# ¡Buen trabajo!

