

# MODELOS AVANZADOS DE COMPUTACIÓN

## PRÁCTICA 2: DATA TYPES

Se pretende realizar un programa que permite seleccionar los 3 ejercicios siguientes y en cada uno de ellos las opciones que se enumeran.

### ENUNCIADO EJERCICIO 1: DROPPRECIO X

Implementar un tipo de datos que almacene una lista de pizzas. Cada pizza a su vez, deberá almacenar:

- Nombre de la pizza
- Los ingredientes
- El precio.

La opción del ejercicio 1 permitirá

1. Listar las pizzas
2. Obtener un listado de las pizzas más baratas de un precio dado
3. Obtener un listado de las pizzas que no tienen el/los ingredientes que se solicitan por teclado.
4. Volver al menú principal

NOTA: en el módulo auxiliar MPizzas se implementaran las funciones necesarias para gestionar la lista de pizzas, tales como `anade_pizza`, `elimina_pizza`, `busca_pizza`....

### ENUNCIADO EJERCICIO 2: GETLIST\_DATE date criterio

Crear un listado de personas empleadas (al menos 5) de una empresa JobVen SL. Cada registro contendrá los siguientes datos

- Dni
- Nombre
- Apellidos
- Fecha de nacimiento
- Cargo
- Fecha de alta

La opción del ejercicio 2 permitirá

1. Listar los empleados con todos sus datos
2. Obtener un listado filtrado por la fecha de nacimiento y/o de alta
3. Buscar por cargo
4. Volver al menú principal

El tipo Fecha tendrá el formato de estar definido de forma que se pueda comparar entre fechas. Almacenará el día, mes y año de nacimiento o alta. El dni, debe almacenar número y letras.

De igual forma, se creará un módulo auxiliar con las funciones necesarias denominado MEmpleados.

## ENUNCIADO EJERCICIO 3: RECORRE ARBOL

Dado el ejercicio anterior, implementar un árbol binario para almacenar las personas y como criterio de ordenación escogeremos la fecha de nacimiento.

1. Listar los empleados utilizando recorrido en profundidad
2. Listar los empleados utilizando recorrido en anchura
3. Buscar un empleado por dni
4. Volver al menú principal

## ENTREGA y MEMORIA

La entrega se realizará a través de la plataforma en un documento .ZIP, el cual debe contener:

- Un fichero Main.hs y los módulos necesarios que se hayan creado (imprescindible)
- Un ejecutable para poder comprobar la ejecución del programa.
- Un documento APELLIDO1-APELLIDO2-NOMBRE\_P3\_Memoria.pdf que deberá contener lo siguiente:
  - Portada
  - Índice
  - Código fuente de cada implementación
  - Descripción del código.

## FECHA LÍMITE DE ENTREGA

Martes 08 de Diciembre a las 23:59

**Nota:** para borrar la pantalla usar el siguiente código.

```
import System.Process
```

```
clear :: IO ()
```

```
clear = system "cls"
```

Nota: para compilar=> `ghc nombreFichero.hs -o nombreFichero.exe`