

# INTRODUCCIÓN A LA PROGRAMACIÓN

## Laboratorio 06

ING. MARTIN POLIOTTO

Docente a cargo del módulo

Septiembre 2020



DIPLOMATURA EN

**NUEVAS  
TECNOLOGÍAS**

SECRETARÍA DE  
EXTENSIÓN  
UNIVERSITARIA

**SEU**

**UTN**  
Facultad Regional Córdoba

Ministerio de  
PROMOCIÓN DEL EMPLEO  
Y DE LA ECONOMÍA FAMILIAR

Ministerio de  
CIENCIA Y  
TECNOLOGÍA

**CBA**  
ENTRE TODOS

Oficina de  
Montevideo  
Organización  
de las Naciones Unidas  
para la Educación,  
la Ciencia y la Cultura

*Practica 01:*

Nos han pedido que desarrollemos un pequeño programa para liquidar los sueldos de una empresa de software de nuestra ciudad. En la empresa trabajan empleados bajo dos modalidades de contratación diferentes:

- Trabajadores que tienen un sueldo fijo pactado al momento de incorporarse a la empresa (generalmente desarrolladores y personal técnico)
- Consultores que trabajan para ciertos proyectos una cantidad variable de horas por mes (generalmente analistas y Project Managers).
- Durante la etapa de análisis del sistema se descubrieron las siguientes entidades: ·

✓ Empleado:

- Trabajador fijo de la empresa. Las propiedades de este objeto son: - legajo - nombre - fecha de contratación-puesto- sueldo- horas extra.
- y debe poseer las siguientes operaciones:
  1. Método calcularPaga(): que retorna el sueldo a percibir en función del sueldo base y las horas extra realizadas en el mes. El sueldo final se calcula como el sueldo base + 300 por cada hora extra hasta 20 horas.
  2. toString(): que retorna los datos completos de un empleado, incluido el valor final de su ingreso mensual.

✓ Consultor:

- trabajador externo que participa en proyectos diferentes. Las propiedades de este objeto son: - legajo - nombre - - fecha de contratación-puesto-cantidad de horas- valor hora
- debe permitir realizar las siguientes operaciones:
  - Método calcularPaga(): que retorna el sueldo a percibir en función de las horas trabajadas entre todos los proyectos en el mes por el valor de la hora pactada.
  - toString(): que retorna los datos completos de un consultor, incluido el valor final de su ingreso mensual.

✓ Empresa:

- Representa la empresa de software. Sus atributos son: cuit (String-13), razón social y empleados (colección de trabajadores de la empresa). Se conoce que actualmente la empresa cuenta con 30 trabajadores, entre empleados fijos y consultores.
- debe poseer las siguientes operaciones:
  - agregar empleado(): agrega el objeto recibido por parámetro a la colección de objetos expuestos
  - incrementarSueldo(porc): permite incrementar el sueldo de los empleados fijos de la empresa en un porcentaje ingresado como argumento.
  - mostrarNomina(): que muestra el listado de los empleados ordenados alfabéticamente.

- `presupuestoMensual()`: que retorna el monto total que la empresa debe asignar en materia de salarios.

### *Practica 02:*

Nos han pedido que desarrollemos un pequeño programa para administrar las fichas de los distintos objetos expuestos en el museo Rocksen. Estos objetos pueden ser de 2 tipos: históricos (piedras, fósiles, plantas, momias, etc.) u obras de arte (esculturas, pinturas, etc.).

Durante la etapa de análisis del sistema se descubrieron las siguientes entidades:

- ✓ Histórico:
  - ✓ elemento antiguo expuesto. Las propiedades de este objeto son: - código - nombre - año: en de creación de elementos naturales - zona: cadena de caracteres que indica la zona geográfica en la que fue encontrado - origen: cadena de caracteres que describe el origen del objeto.
  - ✓ y debe poseer las siguientes operaciones:
    1. Método de acceso nombre: que retorna el nombre del objeto, seguido por la zona, ej: león (África).
    2. Mostrar datos: que retorna todos los datos del objeto en exposición
    3. Método `compareTo()` que compare entre dos objetos históricos cuál es el que tiene mayor antigüedad.
- ✓ Obra:
  - obra de arte expuesta. Las características de este tipo de objetos son: - código - nombre - año: en que fue realizada la obra - autor: nombre del autor de la obra - país: país de origen de la obra
  - debe permitir realizar las siguientes operaciones:

1. Método de acceso nombre: que retorna el nombre de la obra, seguido por su autor, ej: Girasoles 1. (Van Gogh).
2. Mostrar datos: que retorna todos los datos de la obra de arte
3. Método compareTo(): que compare entre dos obras de arte cuál es la que tiene mayor antigüedad.

✓ Sala:

- sala de exposiciones que puede tener cualquier tipo de objeto en exposición y contiene los siguientes atributos: - nombre (generalmente nombradas como Sala A, Sala B, etc.) - ala: ala del museo donde se ubica la sala (Norte o Sur) - objetos: colección de los elementos en exposición
- debe poseer las siguientes operaciones:
  1. agregar objeto: agrega el objeto recibido por parámetro a la colección de objetos expuestos
  2. mostrar datos: muestra por pantalla los datos de todos los objetos en exposición
  3. mostrar la cantidad de objetos en exposición cuya antigüedad es igual o superior a un valor recibido por parámetro.

### Practica 03:

#### Parte I:

Crear una clase Libro que contenga los siguientes atributos:

- ISBN
- Título
- Autor
- Número de páginas
- ✓ Crear sus respectivos métodos get y set correspondientes para cada atributo. Crear el método toString() para mostrar la información relativa al libro con el siguiente formato:

*“El libro con ISBN creado por el autor tiene páginas”*

Crear una clase Adicional Biblioteca que permite modelar nuestro ordenamiento de libros. Agregar los siguientes comportamientos:

- ✓ Un constructor con un parámetro, que permite definir el tamaño de la estantería (cantidad de libros). Validar que la cantidad sea mayor que cero.
- ✓ Un método agregarLibro(unLibro) que permita agregar un libro a la biblioteca siempre que la estantería no se encuentre llena.
- ✓ Un método mostrarListado() que permita retornar el estado de la biblioteca como una cadena:

[Libro1 |Libro2 |Libro3... ]

#### Parte II:

Supongamos ahora que en nuestra biblioteca también almacenamos Revistas. De cada revista se conoce: código, título y número de páginas.

- ✓ Crear el método toString() para mostrar la información relativa a la revista con el siguiente formato:

*“La revista título tiene páginas”*

Refactorizar la clase Biblioteca para que podamos gestionar tanto libros como revistas.

Definir adicionalmente los comportamientos:

- buscarPublicacion(unTitulo) que dado un título permita buscar una publicación en nuestra biblioteca. Si existe mostrar todos su datos. Caso contrario informar con un mensaje.
- contarLibros(nPaginas) que permita conocer la cantidad de libros de la biblioteca con más n páginas, siendo n un parámetro ingresado como parámetro

### *Práctica 04:*

Una empresa de transporte de cargas necesita un software que la ayude a organizarse con la carga de los camiones que maneja. La empresa puede recibir packings, cajas sueltas y bidones que transportan líquido.

- Un **packing** es una estructura de madera que arriba tiene un montón de cajas, se envuelve todo con plástico para que no se desbanden las cajas. Todas las cajas tiene el mismo peso. El peso de un packing es (peso de cada caja \* cantidad de cajas) + peso de la estructura de madera que va abajo. Para cada packing se informa que llevan las cajas, p.ej. Material de construcción.
- De cada **caja** suelta se informa el peso individualmente, son todas distintas.
- El peso de un **bidón** es su capacidad en litros por la densidad (o sea, cuántos kg pesa un litro) del líquido que se le carga. Los bidones van siempre llenos hasta el tope.
- Cada camión puede llevar hasta una carga máxima medida en kg. Además, cada camión puede: estar disponible para la carga (en cuyo caso ya puede tener cosas cargadas), estar en reparación, o estar de viaje.

### **Algunos requerimientos**

- Subir una carga al camión, donde la carga puede ser un packing, una caja suelta, o un bidón. Considerar que no se puede saturar un camión con más peso de lo que su carga máxima permite.
- Bajar una carga del camión, siempre que el camión se encuentre disponible con cargas y que a su vez la carga se encuentre presente dentro de él.



- Conocer el total de cargas de un camión en todo momento y su peso.
- Permitir modificar el estado de un camión, sea porque entro en reparación, salió de reparación, está en viaje o de regreso.
- Obtener el listado de cargas contenidas ordenado por peso.
- Saber si un camión está listo para partir, que es: si está disponible para la carga, y el peso total de lo que tiene cargado es de al menos 75% de su carga máxima.