



1 Material

Para la realización de esta práctica se dispone de los siguientes elementos:

- **Enunciado.pdf:** fichero PDF con este enunciado
- **Fichero de datos:**
 - **inspecciones_restaurantes.csv:** fichero CSV con datos de registros de inspecciones de restaurantes de Nueva York (<https://www.kaggle.com/new-york-city/nyc-inspections>)

2 Datos disponibles

En este proyecto trabajaremos sobre datos de inspecciones de restaurantes en Nueva York. En estos datos encontramos solo un tipo de entidad:

- **Inspección:** contiene información relativa una inspección realizada en un restaurante. Hay que tener en cuenta que en cada restaurante se han realizado una o varias inspecciones.

Los datos están disponibles en formato CSV. En la siguiente figura se muestran las primeras líneas del fichero de datos.

	A	B	C	D	E	F	G	H
1	id	nombre	distrito	tipoCocina	fecha	descripcion	esCritica	score
2	40511702	NOTARO RESTAURANT	MANHATTAN	Italian	06/15/2015	Hot food item not held...	Critical	30.0
3	50046354	VITE BAR	QUEENS	Italian	10/03/2016	Non-food contact surface improperly...	Not Critical	2.0
4	50061389	TACK'S CHINESE TAKE OUT	STATEN_ISLAND	Chinese	05/17/2017	Cold food item held above 41° F...	Critical	46.0
5	41516263	NO QUARTER	BROOKLYN	American	03/30/2017	Live roaches present in facility's food...	Critical	18.0
6	50015855	KABAB HOUSE NYC	QUEENS	Pakistani	03/03/2015	Non-food contact surface improperly...	Not Critical	19.0
7	50058069	HENRI'S BACKYARD	BROOKLYN	American	06/22/2017	Evidence of rats or live rats...	Critical	39.0
8	41547684	PLANET WINGS	STATEN_ISLAND	American	03/10/2015	Cold food item held above 41° F...	Critical	28.0
9	40376944	TOMOE SUSHI	MANHATTAN	Japanese	10/06/2015	Cold food item held above 41° F...	Critical	13.0
10	40560896	YAKITORI TAISHO	MANHATTAN	Japanese	08/13/2015	Plumbing not properly installed...	Not Critical	9.0

3 Modelo

A continuación, se describen todos los elementos que habrá que implementar en este proyecto. Todos ellos se incluirán en el paquete **inspecciones**. Los tipos más destacables del modelo son:

- **Inspeccion**: record para implementar el tipo base.
- **Inspecciones**: tipo contenedor que incluye, además, algunos métodos de consulta basados en tratamientos secuenciales.
- **FactorialInspecciones**: clase para dar soporte a la creación de objetos **Inspeccion** e **Inspecciones** a partir de datos en un fichero CSV.
- **Distrito**: tipo enumerado con los distintos distritos de Nueva York que consideraremos en el análisis.

4 Ejercicios

EJERCICIO 1

Crear el tipo enumerado **Distrito** con los siguientes valores posibles

- MANHATTAN, QUEENS, STATEN_ISLAND, BROOKLYN, BRONX

EJERCICIO 2

Crear el tipo **Inspeccion**, implementándolo como un *record* con las siguientes propiedades

Propiedades:

- **id**: de tipo *String*, es el identificador de la inspección.
- **nombre**: de tipo *String*, representa el nombre del restaurante.
- **distrito**: del tipo enumerado **Distrito**.
- **tipoCocina**: de tipo *String*, es el tipo de cocina.
- **fecha**: de tipo *LocalDate*, representa la fecha de la inspección. La fecha de inspección debe ser posterior al 1/1/1900.
- **descripcion**: de tipo *String*, contiene la descripción del resultado de la inspección.
- **esCritica**: de tipo *Boolean*, permite saber si la inspección ha sido, o no, crítica.
- **score**: de tipo *Double*, es una puntuación numérica de la inspección. El **score** debe ser mayor o igual que cero.
- **tiempoTranscurrido**: de tipo *Period*, es el periodo de tiempo transcurrido entre la fecha de la inspección y la fecha actual.

Constructor:

- A partir de todas sus propiedades básicas, en el orden en el que aparecen en las columnas del fichero.

Representación como cadena:

- Se muestran todas las propiedades básicas del tipo.

Criterio de igualdad:

- Dos objetos de tipo **Inspeccion** son iguales si todas sus propiedades básicas son iguales.

EJERCICIO 3

Crear la clase **Inspecciones** con los siguientes atributos y métodos

Propiedades:

- **inspecciones:** de tipo `List<Inspeccion>`, atributo con una lista de objetos de tipo **Inspeccion**. Consultable.

Constructores:

- C1: constructor sin parámetros, que crea un objeto sin inspecciones.
- C2: constructor que crea un objeto de tipo **Inspecciones** a partir de un *Stream* de **Inspeccion** que se pasa como parámetro.

Representación como cadena:

- Una cadena con todas las inspecciones incluidas en el tipo separadas por un retorno de carro.

Criterio de igualdad:

- Dos objetos de tipo **Inspecciones** son iguales si todas sus propiedades básicas son iguales.

Otras operaciones:

- **Inspecciones::añadirInspeccion:** método para añadir una **Inspeccion** a la lista **inspecciones**.

EJERCICIO 4

Crear la clase **FactorialInspecciones** con los siguientes métodos estáticos

- **FactorialInspecciones::parsearInspeccion:** método privado para construir un objeto **Inspeccion** a partir de una línea CSV del fichero de entrada.
- **FactorialInspecciones::leerInspecciones:** método que devuelve un objeto **Inspecciones** a partir de la ruta del fichero en el que se encuentran los datos de las inspecciones.

EJERCICIO 5

Completar la clase **Inspecciones** con los siguientes métodos

- **Inspecciones::obtenerInspeccionesUltimosAños:** filtra el conjunto inspecciones devolviendo una lista con aquellas de los últimos n años, donde n es un entero recibido como parámetro.
- **Inspecciones::obtenerNombresRestaurantesdeTipoCocina:** devuelve un conjunto ordenado con el nombre de aquellos restaurantes que sirvan un determinado tipo de cocina (que se recibe como parámetro).
- **Inspecciones::calcularMediaScores:** devuelve el valor medio de las puntuaciones de las inspecciones de un determinado restaurante, cuyo nombre se recibe como parámetro.
- **Inspecciones::obtenerRestauranteCritico:** devuelve el nombre de algún restaurante que tenga todas sus inspecciones críticas (o **null** si no lo hubiera).
- **Inspecciones::mostrarInformeInspecciones:** dado el nombre de un restaurante, imprime por pantalla un informe con todas sus inspecciones.
- **Inspecciones::obtenerInspeccionMasReciente:** devuelve la inspección realizada más recientemente.
- **Inspecciones::obtenerInspeccionesEnFechas:** devuelve un conjunto ordenado (por distrito) con las inspecciones realizadas entre dos fechas recibidas como parámetros.
- **Inspecciones::obtenerInspeccionesPorRestaurante:** calcula un diccionario cuyas claves son los nombre de los restaurantes, y el valor es una lista con todas las inspecciones que se han llevado a cabo en él.

- **Inspecciones::obtenerRestauranteMasInspecciones:** devuelve el nombre del restaurante que más inspecciones haya tenido.
- **Inspecciones::obtenerDistritoMasInspeccionesCriticas:** devuelve el distrito con mayor número de inspecciones críticas.
- **Inspecciones::obtenerDescripcionInspeccionMayorPuntuacionPorNombre:** devuelve un diccionario cuyas claves son los nombres de los restaurantes, y el valor es la descripción de la inspección que se haya llevado a cabo en dicho restaurante y haya tenido mayor puntuación.
- **Inspecciones::generarFicheroInspeccionesEnDistritoYFecha:** recibe el nombre de un fichero, un distrito y una fecha y genera un fichero con las inspecciones llevadas a cabo en dicho distrito y fecha.