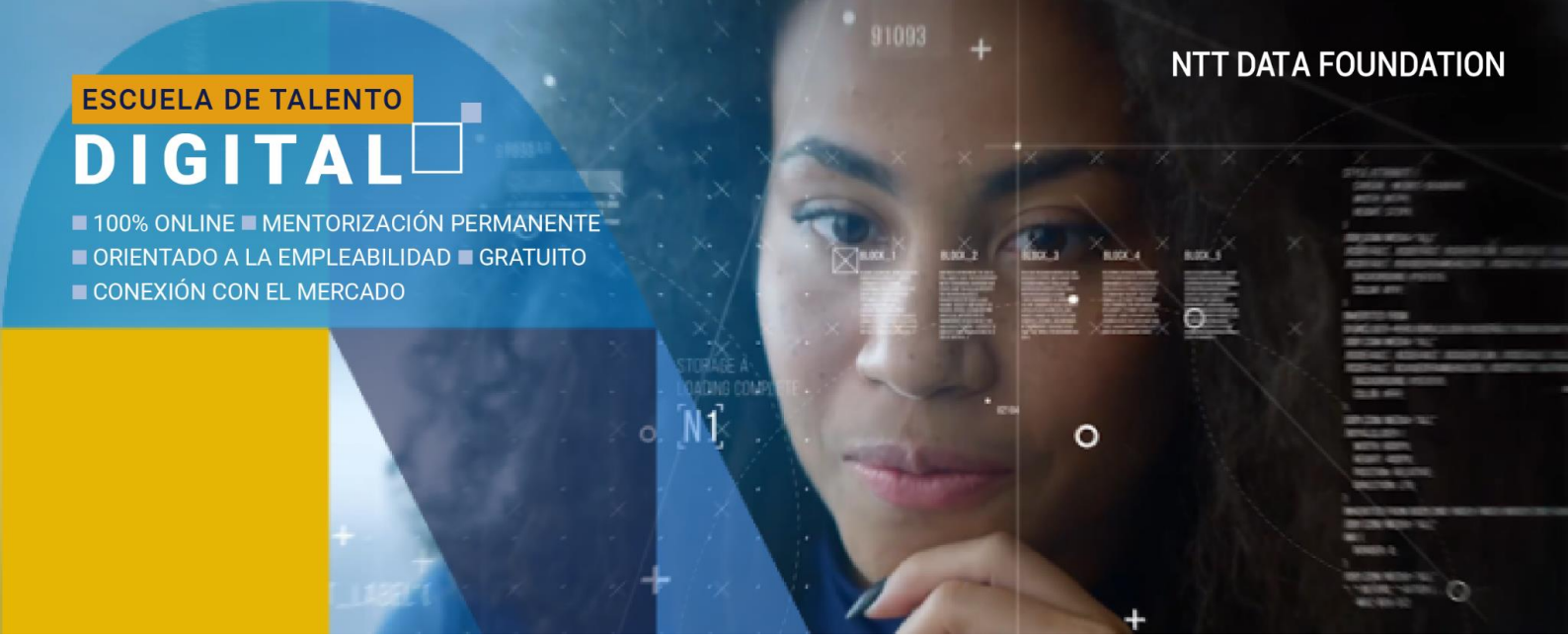


ESCUELA DE TALENTO DIGITAL

NTT DATA FOUNDATION

RETO GRUPAL ÁREA 1

ÍNDICE

1. INTRODUCCIÓN

Para superar este reto grupal, tendréis que ir resolviendo una serie de ejercicios que os vamos a proponer en este documento.

1.1. ¿Cómo entregáis vuestros ejercicios?

Tendréis que preparar un documento pdf y subirlo a la plataforma en el espacio habilitado para ello. No es necesario que todos los componentes del grupo subáis el documento, con que lo suba uno de vosotros es suficiente.

1.2. ¿Qué debe contener el documento pdf?

Este documento deberá contener el código propuesto para resolver cada uno de los ejercicios, pero, además, también debe contener una explicación de cómo habéis llegado a obtener esa solución, que debe ser conjunta y aprobada por todos los miembros del grupo.

Como durante el desarrollo de la actividad van a surgir diferentes propuestas, queremos que las documentéis, es decir, que cuando expliquéis cómo habéis llegado al resultado final, también tenéis que explicar qué otras alternativas había y quién las ha propuesto.

Por ejemplo, imaginad un grupo de 5 alumnos (alumno1, alumno2, alumno3, alumno4 y alumno5) resolviendo el ejercicio 1. La propuesta de resolución del ejercicio 1 debería ser algo como esto:

Después de leer el enunciado, entendimos que lo que se solicitaba era hacer ……

Durante el proceso, el estudiante1 propuso llegar a la solución de la siguiente manera…… pero al estudiante2 y al estudiante3 les pareció mejor hacer …….. y todos estuvimos de acuerdo.

Por todo esto, proponemos esta solución en la que estamos de acuerdo los 5 participantes:

CÓDIGO PROPUESTO

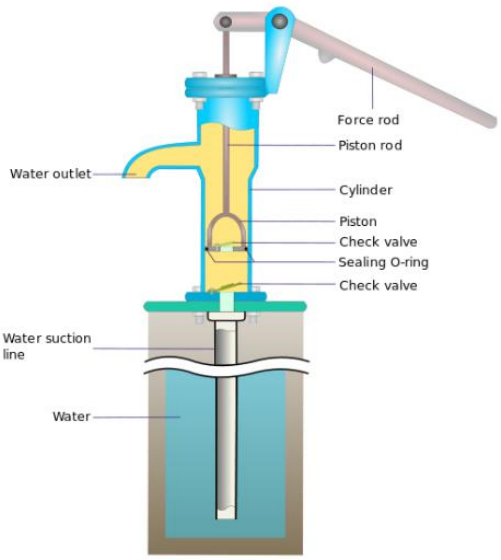
Con esto lo que queremos valorar es la participación de cada uno de vosotros durante el desarrollo del reto.

Cuidad también el formato en el que presentéis el documento, porque también se tendrá en cuenta.

Si tenéis cualquier duda, consultad al tutor a través de la plataforma.

2. CLASIFICACIÓN BINARIA DE PUNTOS DE AGUA, ENTRE FUNCIONALES Y NO FUNCIONALES

Este reto, que desarrollaréis a lo largo de todo el programa, en las diferentes áreas, tiene como objetivo que, al final del curso, logréis predecir qué bombas de Agua de Tanzania funcionan y cuáles no.



La información que se facilitará a lo largo de los distintos retos grupales tiene su base en un dataset obtenido en Driven Data, el cual presenta datos facilitados por el Ministerio de Agua de Tanzania, a cerca del estado de las distintas bombas de agua sobre las que tienen la competencia.

Una comprensión inteligente de qué puntos de agua fallarán puede mejorar las operaciones de mantenimiento y garantizar que las comunidades de Tanzania dispongan de agua limpia y potable.

Además, al finalizar el curso deberéis ser capaces de extraer toda la información posible de los datos facilitados y presentarla de la mejor manera posible, utilizando los gráficos y las herramientas de visualización vistas en clase.

Las variables que contiene este dataset y que, por tanto, servirán para los objetivos descritos anteriormente, son las siguientes:

- amount_tsh – carga estática total (cantidad de agua disponible, para el punto de agua).
- date_recorded –fecha en la que se incluyó el registro en los datos.
- funder –quién financió el pozo.
- gps_height –altitud del pozo.
- installer –organización que lo instaló.
- longitude – coordenada GPS.
- latitude –coordenada GPS.
- wpt_name –nombre del punto de agua, si lo tiene.
- num_private –
- basin –cuenca hidrográfica.
- subvillage –localización geográfica.
- region –localización geográfica.
- region_code –código localización geográfica.
- district_code – código localización geográfica.
- lga –ubicación geográfica.
- ward – ubicación geográfica.
- population –población alrededor del pozo.
- public_meeting – True/False si es punto de reunión.
- recorded_by –grupo que introdujo este registro en los datos.
- scheme_management –quién opera el punto de agua.
- scheme_name –quién opera el punto de agua.
- permit –si el punto de agua está permitido.

- construction_year –año de construcción.
- extraction_type –el tipo de extracción que utiliza el punto de agua.
- extraction_type_group – el tipo de extracción que utiliza el punto de agua.
- extraction_type_class – el tipo de extracción que utiliza el punto de agua.
- management –cómo se gestiona el pozo.
- management_group – cómo se gestiona el pozo.
- payment –coste del agua.
- payment_type – coste del agua.
- water_quality –calidad del agua.
- quality_group – calidad del agua.
- quantity –cantidad de agua que aporta el pozo.
- quantity_group – cantidad de agua que aporta el pozo.
- source –la fuente del agua.
- source_type – la fuente del agua.
- source_class – la fuente del agua.
- waterpoint_type –el tipo de punto de agua.
- waterpoint_type_group – el tipo de punto de agua.

Partiendo de estas premisas, pasamos a enunciar los ejercicios de este primer reto grupal que tendréis que resolver.

¡Nos vamos a Tanzania!

El Ministerio de agua de este país (https://www.maji.go.tz/) ha contactado con nosotros, con el objetivo de lograr reforzar sus sistemas de agua y saneamiento, debido a los inmensos problemas a los que se enfrentan actualmente en este ámbito, más si cabe por el fuerte impacto que está teniendo el cambio climático sobre ellos. Casi el 43% de la población carece de acceso al agua potable básica, y sólo un 25% utiliza un saneamiento gestionado de forma segura.

A día de hoy, más del 70% de las catástrofes naturales del país están relacionadas con el cambio climático y vinculadas a las sequías recurrentes – que han desencadenado la inseguridad alimentaria, la escasez de agua y la falta de electricidad –, así como a las inundaciones – que han provocado el colapso de las infraestructuras de suministro de agua y saneamiento –.

Se calcula que el 60% del PIB de Tanzania está asociado a actividades sensibles al clima. Además, el país gasta el 70% de su presupuesto sanitario en enfermedades prevenibles

relacionadas con el agua, el saneamiento y la higiene, por lo que es fundamental invertir en estos servicios de agua y saneamiento.

Para ayudarnos en nuestra tarea, nos han facilitado un conjunto de datos en el que se detallan las ubicaciones de miles de puntos de agua con los que cuentan, así como otras características de estos, como su año de construcción o la calidad del agua que producen, entre otras.

En este primer reto grupal, pondremos en práctica todo lo aprendido durante el Área 1 del curso, aplicándolo a cierta información extraída del conjunto de datos anterior.

2.1. Ejercicio 1

A continuación, se nos facilita una lista con una serie de identificadores de bombas de agua extraída de la base de datos original:

lista_id_bombas = [8776,34310,67743,19728,9944,19816,54551,53934,46144,49056,50409,36957,50495,53752,61848,48451,58155,18274,48375,6091,37862,51058,55012,9944,20145,19685,69124,46804,6696,12402,41583,57355,67359,60048, 16583,25,70238,12796,52019,19282]

Cread un programa que **elimine los elementos duplicados**, si los hubiera, de la lista anterior y la devuelva libre de ellos. **Comprobad**, midiendo las longitudes de ambas listas (la inicial y la libre de duplicados) **que se ha realizado la limpieza** correctamente.

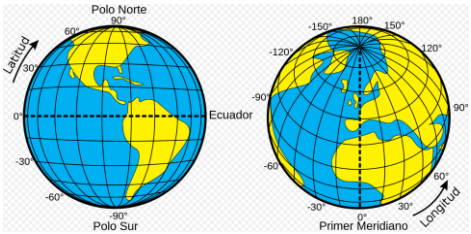
Introducid una nueva id, 59398, dentro de la lista resultante, en la tercera posición.

2.2. Ejercicio 2

Nos informan de que algunos lotes de bombas salieron defectuosos; son, en particular, aquellos que tienen un identificador comprendido entre el 73890 y el 74890. **Cread una función** que solicite que introduzcas un id de alguna bomba y, en el caso de que ese id se encuentre entre los dañados, devuelva el mensaje “La bomba con id “x” se encuentra entre los lotes defectuosos.” En caso contrario, muestra el mensaje correspondiente.

2.3. Ejercicio 3

De todos puntos de agua con los que cuenta nuestra base de datos, nos han facilitado las coordenadas GPS de **latitud** y **longitud** de su ubicación.



Coordenadas geográficas. Fuente: Wikipedia

Por ejemplo, sabemos que el pozo con número identificador 593998 tiene una longitud de 35,861315 y una latitud de -6,378573 (en el sistema estándar decimal), mientras que el pozo con id 593999 tiene longitud 38,104048 y latitud -6,747464. Sabemos que la **distancia entre dos puntos geográficos** puede calcularse a partir de la conocida como **fórmula de Haversine**:

$$d = 2R\,arcsin\left(\sqrt{sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + cos(\varphi_1)\,cos(\varphi_2)\,\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Donde:

- R es el radio de la Tierra, igual a 6372.8 km.
- φ_1 y φ_2, las latitudes (en radianes).
- λ_1 y λ_2, las longitudes correspondientes (en radianes).

a. Teniendo esto en cuenta, **cread** un programa que calcule la **distancia entre los dos pozos de agua** mencionados anteriormente.
b. Ahora, id un paso más allá y **conseguid** (si no lo habéis hecho ya) implementar **una función** que nos solicite introducir las coordenadas de dos puntos cualesquiera, y nos **devuelva la distancia** entre ellos.
c. **Investigad** en pypi https://pypi.org/ si existe ya alguna función que sirva a tal efecto. De ser así, conseguidla y ponedla en práctica.
d. En la siguiente tabla podéis encontrar las coordenadas de otras cinco bombas:

Id	Latitud	Longitud
69572	-9.856322	34.938093
8776	-2.147466	34.698766
34310	-3.821329	37.460664
67743	-11.155298	38.486161
19728	-1.825359	31.130847

e. Utilizando la **librería folium**, **representad** sobre un mapa la posición de las siete fuentes de agua anteriores.

2.4. Ejercicio 4

Se nos ha facilitado un diccionario con el id de una serie de bombas, junto con la fecha en la que se les realizó la última revisión:

fechas_ultima_revision = {"69572":"2023-03-10", "8776":"2023-03-15", "34310":"2023-04-20","67743":"2023-03-02","19728":"2024-02-08", "9944":"2024-05-04","19816":"2023-07-

07","54551":"2023-02-22","53934":"2023-01-05","46144":"2024-05-10", "49056":"2023-06-17","50409":"2023-02-13","36957":"2023-08-02","50495":"2023-05-23","53752":"2023-03-01"}

Teniendo en cuenta que la recomendación es que se realice una revisión anual a cada punto de agua, **extraed**, por un lado, aquellas bombas que, a día de hoy, **tienen las revisiones en orden** y, por otro, aquellas que ya **deberían haber sido revisadas, junto con su fecha límite**.

Una posible manera de abordar este problema sería la siguiente:

- **Cread** un bucle que transforme todas las fechas dadas a un formato correcto.
- **Sumad** a cada una de ellas 365 días (se correspondería con la fecha en la que ya se tendría que haber revisado la bomba de agua).
- **Comparad** las fechas anteriores con el día de hoy.
- Si dicha fecha es menor que “hoy”, la bomba correspondiente ya debería haber sido revisada. Si no, es decir, si la fecha es mayor que “hoy”, aún está dentro del plazo estimado.

Para hacer esto, serían necesarias algunas funciones de la librería datetime (today(), strptime(), timedelta(),.date(),…).

El resultado debería tener un aspecto similar al siguiente (teniendo en cuenta que estas pruebas las realizamos el 30/01/2023):

```
Las bombas que aún tienen la revisión pendiente, junto con la fecha de revisión señalada, son las siguientes{'69572': '2023-01-10', '8776': '2023-01-15', '67743': '2023-01-02', '53934': '2023-01-05'}
Las bombas que aún están en fecha son{'34310': '2023-02-20', '19728': '2023-02-08', '9944': '2023-04-04', '19816': '2023-03-07', '54551': '2023-02-22', '46144': '2023-05-10', '49056': '2023-03-17', '50409': '2023-02-13', '36957': '2023-03-02', '50495': '2023-05-23', '53752': '2023-04-01'}
```

Ahora, con la lista de bombas obtenidas que tienen la revisión pendiente, con plazo expirado, **ordenadlas** de manera **ascendente**, de tal manera que la primera sea la que más retraso acumula.

Por último, **mostrad** por pantalla el **id** de la bomba que más retraso acumulado lleva para su revisión. Sería algo como lo siguiente:

```
El id de la bomba cuya revisión más retraso acumulado tiene es: 67743
```