

Máster Universitario en Ingeniería Industrial

Trabajo Fin de Máster

Diseño e implementación de un sistema autónomo para la detección de fugas en depósitos

Autor/a: Nicolás Yanes Pérez

Tutor/a: Pedro Antonio Toledo Delgado

Cotutor/a : Marta Sigut Saavedra

7 de Septiembre de 2022

La publicación de este Trabajo Fin de Máster sólo implica que el estudiante ha obtenido al menos la nota mínima exigida para superar la asignatura correspondiente, no presupone que su contenido sea correcto, aunque si aplicable. En este sentido, la ULL no posee ningún tipo de responsabilidad hacia terceros por la aplicación total o parcial de los resultados obtenidos en este trabajo. También pone en conocimiento del lector que, según la ley de protección intelectual, los resultados son propiedad intelectual del alumno, siempre y cuando se haya procedido a los registros de propiedad intelectual o solicitud de patentes correspondientes con fecha anterior a su publicación.

Índice

RESUMEN.....	6
ABSTRACT.....	7
1. CONSIDERACIONES INICIALES.....	8
1.1. Antecedentes del proyecto y objetivos.....	8
1.2. Project background and objectives.....	10
1.3. Esquemas del desarrollo del proyecto.....	12
1.4. Reglamentos y normas de aplicación.....	13
2. FUGAS DE COMBUSTIBLE EN LAS ESTACIONES DE SERVICIO.....	14
2.1. Almacenamiento de hidrocarburos en una estación de servicios.....	14
2.2. El peligro de las fugas en los depósitos de estación de servicios.....	17
2.3. Causas de fugas en los depósitos de almacenamiento de combustible.....	18
2.4. Sistemas de detección de fugas en estaciones de servicio.....	19
3. SIMULADOR DE UNA PLANTA DE DEPÓSITOS DE COMBUSTIBLE.....	20
3.1. Antecedentes.....	20
3.2. Metodología de la programación del simulador.....	21
3.3. Diseño del simulador de la planta.....	22
3.3. Funcionamiento del simulador de la planta.....	30
3.4. Datos obtenidos.....	33
4. DETECCIÓN FUGAS POR MEDIO MACHINE LEARNING.....	35
4.1. Aplicación de Machine Learning en la detección de fugas.....	35
4.2. Desarrollo del sistema de detección fugas.....	38
4.3. Resultados obtenidos.....	43
5. CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO.....	46
5.1. Conclusiones.....	46
5.2. Conclusions.....	47
5.3. Mejoras y nuevas líneas de trabajo.....	48
ANEXOS.....	49
Anexo 1. Presupuesto.....	49
Anexo 2. Guía del desarrollador.....	51
Anexo 3. Guía del usuario.....	54
REFERENCIAS.....	56

Índice de figuras

Figura 1. Investigación de la Guardia Civil en la gasolinera de Coria del Río (Sevilla).....	8
Figura 2. Esquema del proyecto.....	12
Figura 3. Elementos principales de suministro en una estación de servicio.....	14
Figura 4. Depósito de simple pared dentro de una cubeta de hormigón.....	15
Figura 5. Depósito de doble pared.....	15
Figura 6. Bomba surtidor combustible gasoil.....	16
Figura 7. Río Jarama (Madrid), afectado por una gran fuga de vertidos y combustible.....	17
Figura 8. Revestimiento interno en fibra de vidrio en tanque de almacenamiento de combustible...	18
Figura 9: Simulación con arduinos por medio de la wep app Tinkercad.....	20
Figura 10. Esquema de la planta de la simulación.....	24
Figura 11: Máquina de estados de un depósito activo.....	25
Figura 12: Distribución normal.....	27
Figura 13. Máquina de estados de la controladora.....	28
Figura 14: Esquema de la clase planta.....	30
Figura 15: Representación de los datos del volumen de los depósitos del simulador.....	32
Figura 16. Esquema de entrenamiento de un modelo de clasificación multiclase.....	36
Figura 17. Ejemplo de la ventana de tiempo.....	37
Figura 18. Esquema del sistema de detección de fugas aplicando ML.....	38
Figura 19. Gráfica Curva ROC.....	41
Figura 20. Validación cruzada k-folds (K=5).....	43
Figura 21. Resultados del modelo con un inventario de un año.....	44
Figura 22: Resultados del modelo con un inventario mensual.....	45

Índice de tablas

Tabla 1. Fragmento de tabla de datos de la simulación de la estación de servicios.....	33
Tabla 2. Matriz de confusión.....	40
Tabla 3. Presupuesto de un equipo informático para el desarrollo del proyecto.....	49
Tabla 4. Presupuesto correspondiente al trabajo de desarrollo del proyecto.....	50

Índice de ecuaciones

Ecuación 1. Caudal variable de fugas.....	26
Ecuación 2. Métrica Accuracy.....	40
Ecuación 3. Métrica Precisión.....	40
Ecuación 4. Métrica Recall.....	40
Ecuación 5. Métrica F1 Score.....	40

RESUMEN

Un aspecto clave en el almacenamiento de hidrocarburos en estaciones de servicios es la detección de fugas en las mismas para evitar propagar sustancias contaminantes que puede dañar el medio ambiente y a las poblaciones circundantes. A causa de esto es interesante investigar nuevas formas más eficientes para la detección de fugas.

Sin embargo, no son muchas las estaciones de servicios que deciden compartir sus libros de inventario de combustible y la cantidad de muestras que se disponen es bastante limitada. Para solventar esto, en este proyecto, se pretende crear un simulador de una planta prototipo que emule el comportamiento de una estación de servicio para obtener datos para el estudio de detección de fugas. Para ello, se tomó como referencia el diseño de la planta del trabajo final de grado del alumno del grado de Ingeniería Electrónica Industrial y Automática Luis Arriaga Campos “*Diseño e implementación de un sistema autónomo para la simulación de fugas en depósitos*” [10] que emula el comportamiento de una estación de servicios. Y a partir de este diseño, programamos un simulador, utilizando lenguaje de programación python cuyo resultado nos permite generar datos simulados de un inventario de una estación de servicios. Este simulador supondrá una ventaja para futuras investigaciones en técnicas de detección de fugas pues permitirá realizar múltiples pruebas sin a priori depender de empresas externas que compartan sus datos o del uso prototipos de estaciones, con el supuesto coste que suponen.

Una vez obtenidos las muestras de datos simuladas y basándonos en anteriores estudios en los que participan los tutores de este proyecto [5][6], se desarrolla un sistema de detección de fugas utilizando modelos de clasificación aplicando *Machine Learning*. El sistema seleccionará cual es el mejor modelo en base a sus resultados de detección y posteriormente se les introducimos nuevas muestras de datos, aportándonos buenos resultados y cumpliendo con la normativa vigente en relación a la detección de fugas [1].

ABSTRACT

In the storage of hydrocarbons in petrol stations, it is important to detect fuel leaks to avoid spreading polluting substances that can damage the environment and cause health problems for the population. Because of these, it is interesting to investigate new, more efficient forms of leak detection.

However, petrol stations are very reluctant to share their fuel inventories and therefore there is not enough storage of fuels data. To solve this, in this project it is intended to create a simulation of the storage of a gas simulator. For this, we take as a reference the design of the plant of the final degree project by Luis Arriaga Campos "*Design and implementation of an autonomous system for the simulation of leaks in tanks*" [10] and using the programming language python, we program this simulator that allows to generate simulated data of a fuel inventory of a petrol station. This simulator will be an advantage for future investigations based on the detection of tank leaks as it allows multiple tests to be carried out without the need to depend on external companies that share their data or the use of petrol station prototypes and their costs.

With the simulated data and using previous studies in which the tutors of this project participate [5][6], A leak detection system is developed using classifier models through *Machine learning* methods. And based on these prediction models and feeding them with new data samples, we managed to obtain a high detection efficiency and comply with current regulations in relation to leak detection [1].

1. CONSIDERACIONES INICIALES

1.1. Antecedentes del proyecto y objetivos

El uso de hidrocarburos en la actualidad sigue siendo una fuente de energía esencial para el funcionamiento y desarrollo de nuestra sociedad a pesar de los problemas contaminantes que generan. Y para su consumo es necesario disponer de puntos de ventas como las gasolineras o estaciones de servicio que se reparten por el territorio para que los usuarios puedan abastecerse.

La particularidad de estos puntos de suministro es que requieren del almacenamiento subterráneos para los hidrocarburos, y si no se toman las medidas de seguridad adecuadas o si no existe un mantenimiento y una monitorización de las fugas pueden producirse emisiones de hidrocarburos en el medio que se sitúa y causar daños medioambientales y sanitarios.



Figura 1. Investigación de la Guardia Civil en la gasolinera de Coria del Río (Sevilla)

Ante la importancia y el interés común en la detección de fugas de estos depósitos de almacenamiento de combustibles, un grupo de investigadores del departamento de Ingeniería Informática y de Sistemas de la Universidad de la Laguna (en los que participan los tutores de

este trabajo) han realizado investigaciones basadas en la aplicación de técnicas de clasificación para la detección de fugas utilizando como datos de prueba libros de inventarios de estaciones de servicios reales [5][6]. Estos libros de inventario contienen información de los movimientos de combustibles como la descarga de cubas en el depósito de almacenamiento de la estación, las ventas suministradas en los surtidores, el volumen del depósito de almacenamiento al inicio y al final de día etc.

Depender de los libros de inventario de estaciones de servicios reales, conlleva una serie de inconvenientes :

- Depender de la buena voluntad de las empresas de compartir sus libros de registros para las investigaciones.
- La frecuencia con la que se registran los datos en los libros, ya que el intervalo en la toma de medidas puede llegar a ser incluso mayor del día.
- Pueden contener errores por parte de los empleados o incluso datos desfasados en el tiempo. Un ejemplo sería que anotaran la descarga de una cuba al día siguiente que sucedió.

Ante estos problemas, se realizó en 2020 en la Universidad de La Laguna un trabajo final de Grado centrado en diseñar y construir un prototipo físico de una estación de servicios que permitiría solventar los problemas anteriormente citados. Lamentablemente, no se pudo llevar a cabo la instalación física del prototipo a causa de las limitaciones de acceso a las instalaciones de la universidad debido a la pandemia por el COVID19 [10]. Por lo que, el autor implementó como alternativa una simulación de esta planta a través de dos microcontroladores arduinos y un software web de simulación de circuitos.

En base a este diseño de prototipo de estación de servicios, se pretende mejorar la implementación de la simulación del prototipo del trabajo citado y además aplicar técnicas de *Machine Learning* para la detección de fugas partiendo de las investigaciones anteriormente mencionadas y utilizando los datos generados por la simulación.

Por tanto se establecen los siguientes objetivos a cumplir:

1. Obtener datos simulados de un libro de inventario de una estación de servicios.
2. Con los datos obtenidos, aplicar técnicas de *Machine Learning* que detecten si existen fugas en depósitos de almacenamiento de hidrocarburos o no.

Para cumplir estos objetivos se desarrollará :

- Una herramienta de simulación que nos permita obtener datos simulados de un prototipo de estación de servicios con las características que se requieran y que no dependa de licencias de software externos privativas.
- Un sistema de *Machine Learning* autónomo que establezca de forma óptima, el mejor modelo de detección de fugas partiendo de los datos que se obtengan de la simulación.

1.2. Project background and objectives

Currently, the use of hydrocarbons continues to be an essential energy resource for the functioning of our society, despite the pollution problems they generate. In order for users to be able to consume these fuels, it is necessary to have sales and distribution points throughout the territory, the well-known petrol stations.

The peculiarity of these hydrocarbon sales points is that they require the fuel underground storage, and if safety measures are not taken seriously or if there is no exhaustive control and maintenance, hydrocarbon emissions can occur in the surrounding environment and it could cause environmental and health damages.

Given the importance and common interest in detecting leaks in these fuel storage tanks, groups of researchers from the Department of Computer and Systems Engineering of the University of La Laguna (in which the tutors of this work participate) have carried out investigations in this regard. As a result, two publications were made in the JCR magazine (Journal Citations Report) based on the application of classification techniques for leak detection using real petrol station inventory books as test data. These inventory books

contained information on fuel movements such as the filling of fuel storage tank, the sales, etc.

Relying on petrol station inventory books, it had a number of disadvantages:

- Rely on the willingness of companies to share their inventory books for investigations.
- The lack of data in the inventory books, since the interval time measurements could be more than one day.
- They may contain errors caused by employees or even data that is out of date with time. An example would be that they recorded the discharge of a tank the day after it happened.

Faced by these problems, In 2020, a final degree project was developed at the University of La Laguna to design and build a physical prototype of a petrol station that would solve the problems previously mentioned. Unfortunately, the installation of the prototype could not be carried out because of the access limitations of the university facilities due to the COVID19 pandemic. Therefore, the author implemented a simulation of this plant, using two arduino microcontrollers and a web-based circuit simulation software.

Based on the design of the prototype, in this master's degree dissertation it is intended to improve the implementation of the simulation of the prototype of the final degree work and also to apply Machine Learning leak detection techniques based on the research previously mentioned.

Therefore, the following objectives are established:

1. To develop a simulator of the designed prototype that allows get simulated data from a petrol station.
2. With this data, we apply Machine Learning techniques that detect tank leakage.

To achieve these objectives it is necessary:

- A simulation tool that allows us to obtain simulated data from the prototype of a petrol station with the characteristics required for the investigation of leak detection, at no cost and it must not depend on third proprietary software.
- Develop an autonomous *Machine Learning* system to establish the best leak detection model based on the data obtained from the simulation

1.3. Esquemas del desarrollo del proyecto

El proyecto se puede descomponer en los siguientes elementos principales:

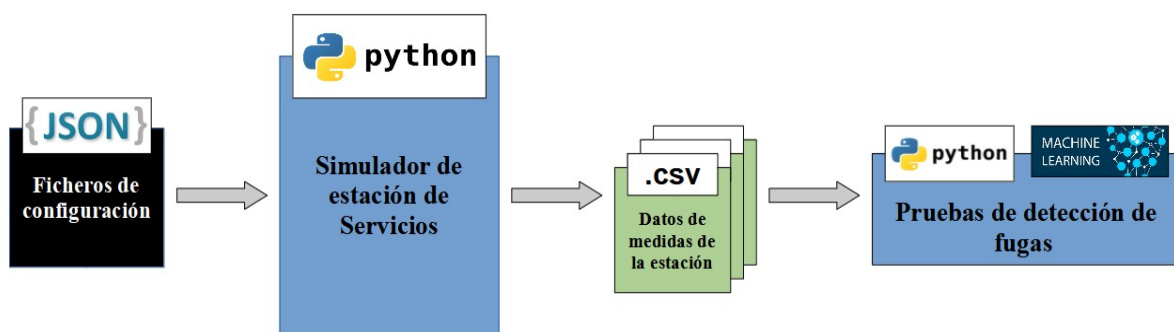


Figura 2. Esquema del proyecto

- Ficheros de configuración: son ficheros que podrán editar los usuarios para establecer distintos parámetros de la simulación, como el tiempo de simulación, el intervalo de toma de medidas o definir características de los depósitos simulados. Estos ficheros tienen que editarse en base al formato de texto JSON.
- El simulador de planta de una estación de servicios : para este simulador se establece como lenguaje de programación python por su versatilidad y su fácil uso con respecto a otros lenguajes. El usuario final no tendrá que editar los ficheros internos del simulador para cambiar parámetros de la simulación. Únicamente deberá de ejecutar el script de python (`sim_plant.py`) encargado de realizar la simulación, tomando automáticamente la configuración de los ficheros JSON.
- Generación del libro de inventario de la estación simulada: Como resultado de la simulación obtendremos una lista de datos de medidas diarios de la simulación de la planta, basados en los registros de inventarios de las estaciones de servicios reales de

las investigaciones anteriormente mencionadas [5][6]. El formato utilizado para montar estos registros es el CSV.

- Sistema de detección de fugas: este sistema estará programado en python ya que este lenguaje de programación dispone de librerías dedicados al *Machine Learning* (ML). Se desarrolla con el objetivo que aporte el mejor modelo de identificación de fugas a partir de los datos simulados de la estación de servicios.

1.4. Reglamentos y normas de aplicación

En el desarrollo del proyecto es necesario tener en cuenta la normativa vigente en relación a la detección de fugas de combustible en estaciones de servicio. Los documentos a considerar son los siguientes

- UNE-EN 13160-5, Sistemas de detección de fugas. Parte 5: Requisitos y métodos de ensayo/evaluación de los sistemas de detección de fugas en tanques con indicador de nivel y en los sistemas de tuberías a presión [1].
- Real Decreto 706/2017, de 7 de julio, por el que se aprueba la instrucción técnica complementaria MI-IP 04 "Instalaciones para suministro a vehículos" y se regulan determinados aspectos de la reglamentación de instalaciones petrolíferas [2].
- Real Decreto 2085/1994, de 20 de octubre, por el que se aprueba el Reglamento de Instalaciones Petrolíferas [3].

En la elaboración de de la memoria del documento también se ha respetado las directrices de formato y composición detallados en el siguiente reglamento :

- Reglamento y normas de presentación de los trabajos fin de máster. Máster en ingeniería industrial. Escuela de doctorado y estudios de postgrado. Universidad de la Laguna. (17 de julio de 2019) [4].

2. FUGAS DE COMBUSTIBLE EN LAS ESTACIONES DE SERVICIO

2.1. Almacenamiento de hidrocarburos en una estación de servicios

Las estaciones de servicios o gasolineras son conocidas por ser puntos de venta y suministro de hidrocarburos para los vehículos de motor. Obligatoriamente los depósitos donde se almacenan los combustibles se deben ubicar de forma subterránea para evitar el contacto con el exterior y disminuir el peligro de rotura, además de ser más seguro para los propios usuarios de la estación. También se debe evitar el contacto directo entre los distintos hidrocarburos que ofrece la estación por lo que se suele disponer como mínimo de un depósito de almacenamiento por cada tipo de producto a la venta [8].

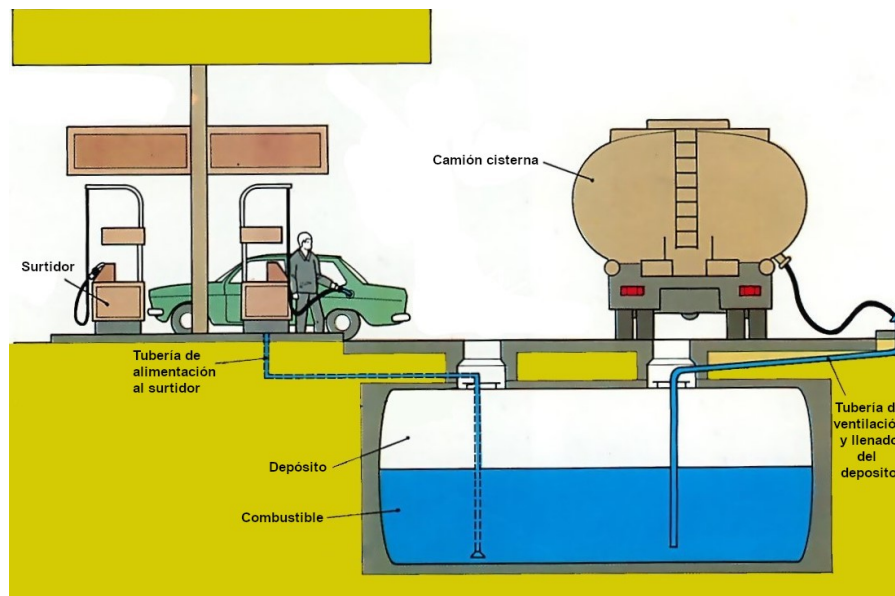


Figura 3. Elementos principales de suministro en una estación de servicio

En la actualidad hay dos tipos de depósitos que se utilizan: de simple pared o de doble pared.

2.1.1. Depósitos de simple pared

Son depósitos fabricados de una capa de plástico reforzado o acero, que se se entierran en un foso común o en un cubeto estanco de hormigón armado para contener el combustible en caso

de fuga. Para detectar posibles fugas se utiliza un tubo vertical que permite ver si hay combustible fugado dentro de la cubeta.

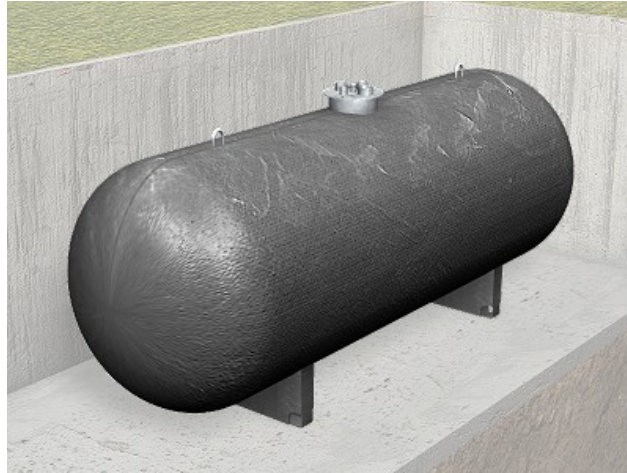


Figura 4. Depósito de simple pared dentro de una cubeta de hormigón

2.1.2. Depósito de doble pared

Este tipo de depósitos están contruidos de tal forma que tienen una doble capa presentando un espacio vacío entre ellas, lo que reduce la posibilidad de fuga. No obstante, la normativa les exige el uso de un sistema de detección de fugas ante la eventualidad de que se produzcan grietas.

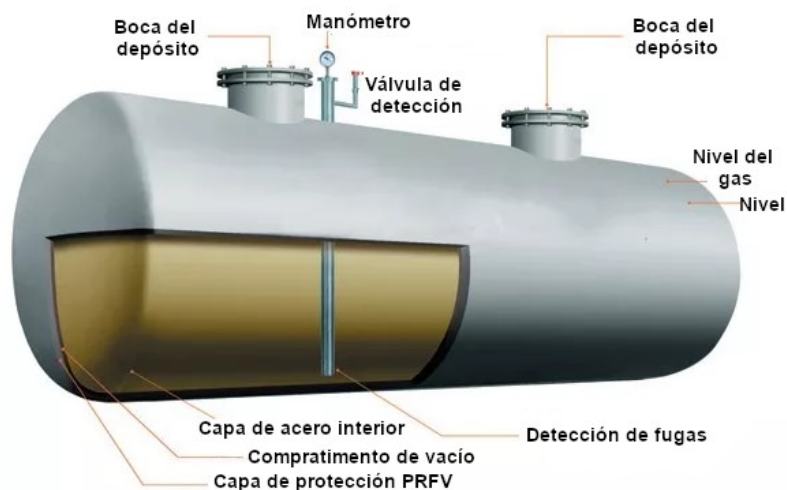


Figura 5. Depósito de doble pared

2.1.3. Surtidor

Para la extracción de líquido de los depósitos subterráneos y suministrar al punto de venta o surtidor donde está repostando el vehículo, es necesario el uso de bombas. Se suelen emplear las bombas de succión ubicadas por encima del depósito de almacenamiento y provocando el ascenso del hidrocarburo por la tubería del dispensador, a través de la disminución de la presión reduciendo el aire de la tubería. Cuando la bomba de succión se apaga después del suministro, se evita que el combustible regrese al depósito de almacenamiento por medio de una válvula de retención. Otra tipo de bombas que se utilizan son las sumergibles dentro del propio tanque de almacenamiento, que a través de una hélice giratoria hace que el combustible suba hasta el surtidor.

Para el control de descarga del depósito de almacenamiento solicitado por el surtidor, se dispone de un medidor de flujo que cuenta con una membrana de plástico que se comprime en base a la cantidad límite combustible pedida y esté se comprime completamente impidiendo la salida de combustible una vez alcanzado el límite.



Figura 6. Bomba surtidor combustible gasoil

2.2. El peligro de las fugas en los depósitos de estación de servicios

Más allá de las pérdidas económicas causadas por las fugas, la principal razón por la que es tan importante su detección en los depósitos de almacenamiento es a causa de los daños medio ambientales que pueden producir debido a la alta toxicidad de los hidrocarburos que se almacena. Las fugas de este tipo de sustancias y su absorción por el subsuelo puede contaminar acuíferos, ríos o aguas subterráneas de consumo humano que puede derivar a problemas en la salud de las poblaciones colindantes.

Además pueden expulsar gases tóxicos al exterior, contaminando el aire cercano a la fuga. Varios de estos gases son difíciles de detectar y son nocivos tanto para la atmósfera como para los seres vivos cercanos a la zona de la fuga.



Figura 7. Río Jarama (Madrid), afectado por una gran fuga de vertidos y combustible

La reparación de estos daños puede implicar grandes costes económicos y en caso de detectar fugas en los depósitos y no tomar medidas, puede implicar imputaciones penales [7].

2.3. Causas de fugas en los depósitos de almacenamiento de combustible

La principal causa que provoca la fuga de los tanques de almacenamiento de los líquidos es la aparición de corrosión en las paredes del depósito provocando picaduras en el acero. Por ello, los depósitos suelen por defecto llevar una capa protectora pasiva de la corrosión (suele usarse plástico reforzado con vidrio). Lamentablemente, no es suficiente ya que puede producir ligeros desperfectos en el aislamiento que finalmente generen en zonas de pequeño tamaño picaduras y por tanto fugas que son muy difíciles de identificar. Este tipo de desperfectos puede originarse durante el transporte o en la propia instalación en la estación de servicios [9] [10].



Figura 8. Revestimiento interno en fibra de vidrio en tanque de almacenamiento de combustible

Las capas protectoras de los tanques además, son percederas ya que no se pueden asegurar su eficacia ante la corrosión en un tiempo limitado de en torno a los 10 años. Por lo que para alargar la vida de estos tanques son necesarios aplicar un mantenimiento adecuado y además implementarse métodos de protección activa que permitan identificar las fugas, tales como la monitorización de parámetros de almacenamiento como los volúmenes de los tanques o el uso de dispositivos diseñados para la identificar la fugas de depósitos.

2.4. Sistemas de detección de fugas en estaciones de servicio

Exponemos una serie de métodos que se utilizan en la actualidad para la identificación de fugas en los depósitos de doble pared de almacenamiento de hidrocarburos [10].

2.4.1. Detección de fugas por monitorización y control de existencias

Es posible detectar cuando se produce una fuga por medio de la monitorización en el depósito por medio de sensores en el interior del tanque y en las cámaras intermedias tomando medidas del nivel de tanque, densidades, consumo por hora, presiones y últimas operaciones de descarga, etc. También se puede visualizar la cámara intersticial sin tener que abrir el depósito. Generalmente, también se dispone de indicadores sonoras y luminosas para alertar al personal de la estación de servicio.

2.4.2. Detección de fugas por medio líquido eléctricamente conductor situado entre las dos paredes del depósito

Este método utiliza un electrodo que nos permite saber la cantidad de líquido intersticial que presenta el tanque. Con este sistema somos capaces de detectar inmediatamente una fuga, pero tiene el gran inconveniente de que se transmite electricidad al depósito, por lo que se suele descartar la instalación de este tipo de sistema por los peligros que conlleva.

2.4.3. Detección de fugas por vacío

Este sistema por medio del control del vacío que se encuentra entre las dos paredes del tanque puede identificar fugas en la capa. A través de microcontroladores puede ayudar a identificar una fuga de una pérdida de vacío ocasional. Además, este sistema tiene la ventaja que nos permite inspeccionar la cámara sin tener que abrir el depósito.

3. SIMULADOR DE UNA PLANTA DE DEPÓSITOS DE COMBUSTIBLE

3.1. Antecedentes

Como ya se comentó en el apartado 1.1 para el desarrollo de este TFM partimos de un TFG previo [10]. En dicho trabajo se llevó a cabo una simulación del comportamiento de una estación de servicios que no pudo construirse, pero comprobó su funcionamiento por medio de la implementación de dos microcontroladores Arduinos y un software de simulación electrónica, Tinkercad.

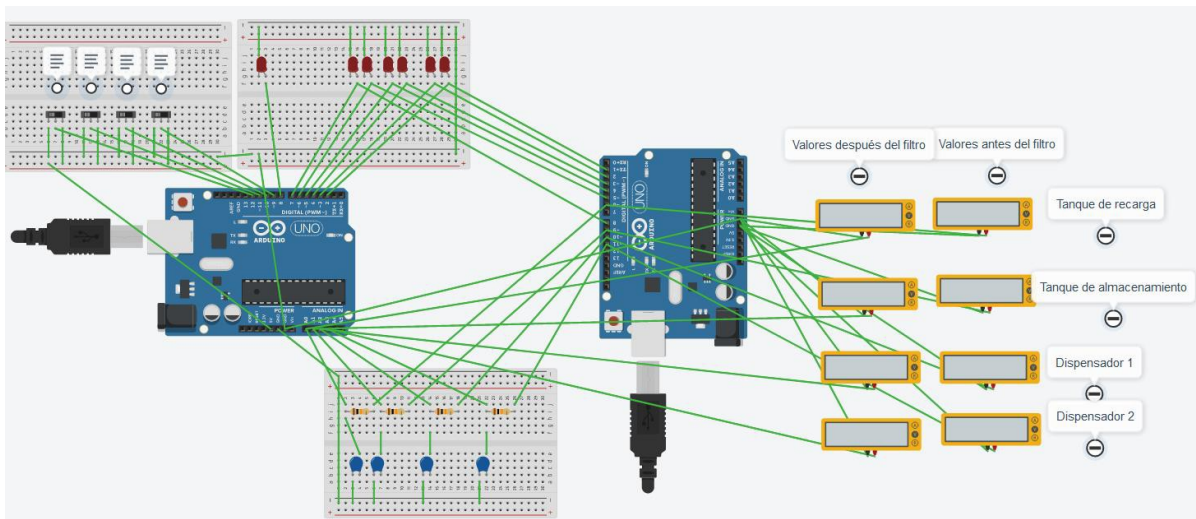


Figura 9: Simulación con arduinos por medio de la web app Tinkercad

Esta implementación conectaba entre sí los dos microcontroladores, actuando uno de ellos como controlador de la planta prototipo, mientras que el otro microcontrolador simulaba las señales que debía de emitir la planta real, comportándose como una especie de simulador.

La principal razón por la que se decidió pasar de este simulador a uno completamente desarrollado por python fue por la versatilidad que aporta tener tanto la parte del simulador como las pruebas de *Machine Learning* en un único lenguaje y sin depender de software externo privativo. Además de poder tener un mayor control en la simulación y en la obtención de datos.

Cómo se tenía acceso al código de los microcontroladores programados [24], se planteó inicialmente hacer un ‘refactoring’ del código original para facilitar la creación del simulador, es decir, conservar el funcionamiento original del código pero cambiando de lenguaje de programación de C a Python. No obstante durante el desarrollo de refactoring del código de los arduinos a python se tuvo que cambiar muchos aspectos del código porque la conversión a lenguaje python respetando su método de programación requería de bastante tiempo y quedaba un código más enrevesado y poco eficiente. Por lo que finalmente se descartó la idea de intentar respetar completamente el método utilizado en el trabajo. Sino recoger ciertos elementos del código y a partir de estos desarrollar un nuevo simulador con una perspectiva similar, pero teniendo en cuenta el uso del nuevo lenguaje utilizado.

3.2. Metodología de la programación del simulador

3.2.1. Programación orientada a objetos

Para el desarrollo del simulador se ha seguido la metodología de programación orientada a objetos ó OOP. Esto implica que el desarrollo gira en torno a la creación de objetos que serán los elementos a manipular y que permitan realizar acciones dentro del programa. Este enfoque de programación es adecuado sobre todo para programas de gran tamaño o porque su metodología facilita la comprensión del código que se desea implementar [12]. En este caso, tiene sentido pues nos permite identificar y manipular elementos que puedes encontrar en una planta prototipo de una estación de servicios real. A causa de esto, se desarrollaron las siguientes clases (molde para crear el objeto) para implementar el simulador:

- Depósito (tank.py).
- Tubería (pipe.py).
- Caudal (flow.py).
- Evento (event.py).
- Ruido (Noise.py).
- Planta (plant.py).
- Controladora (controller.py).

En los posteriores apartados, explicamos su implementación y las funciones que realizan dentro del simulador.

No obstante previamente, se debe mencionar otros aspecto en la modelación del simulador.

3.2.2. Combinación de simulación continua y simulación de eventos discretos

Para el diseño de modelación del sistema de la planta se estableció una perspectiva de simulación continua, es decir, con variables de estado siendo actualizadas en el tiempo según la dinámica del sistema, como por ejemplo la variable volumen de los tanques. Sin embargo, hay partes del simulador cuyo comportamiento se basa en eventos discretos. Esto implica que hay variables de estado que no cambian hasta que suceden ciertos eventos de instantes de tiempos separados. Esto sucede, por ejemplo, en el caso de la controladora que determinado su estado en base a determinados eventos.

3.3. Diseño del simulador de la planta

Como se ha mencionado anteriormente, el diseño de la planta que simula una estación de servicios, procede del TFG citado previamente [10]. En ese trabajo, la planta prototipo que se deseaba construir estaba constituido por una serie de tanques que emulaban los depósitos de una estación conectados por tuberías, cómo se puede ver en la figura 10. Para la implementación de este diseño fue necesario definir dos tipos de objetos fundamentales en la dinámica del sistema, los depósitos y las tuberías.

3.2.1. Depósitos

Los objetos depósitos almacenan los registros relacionados con esté elemento que representan, es decir, atributos como su capacidad, su límite de volumen de seguridad, la cantidad de litros de combustible que contiene, etc. Y se distinguen dos tipos:

- Los tanques activos: son depósitos que tienen acceso y control de sus tuberías conectadas. Lo que les aporta la capacidad de controlar en la simulación la cantidad de

- volumen que recibe o aporta a los depósitos que se encuentran conectados. Dentro de la planta se identifican los siguientes tanques:
- Depósito de llenado: este depósito representaría la cuba del camión cisterna que llena el depósito de nuevo combustible.
 - Surtidor 1 y 2: representaría depósitos ficticios donde va el combustible comprado por los consumidores de la estación.
 - Depósito de fugas: depósito que almacenaría el líquido fugado del depósito de almacenamiento para las pruebas de la planta. En la simulación la fuga se simula a través de una tubería
- Los tanques pasivos: son depósitos que recibe o se le extrae líquido pero no tiene la capacidad de manipular el líquido de sus depósitos adyacentes conectados.
 - Depósito de almacenamiento. Corresponden al depósito de almacenamiento de la estación y del cual estudiaremos el caso de fuga.
 - Depósitos de retorno. Este tanque recogería el líquido de los dispensadores y del depósito de la fuga para cuando se necesite de nuevo enviar líquido al depósito de llenado en la simulación.

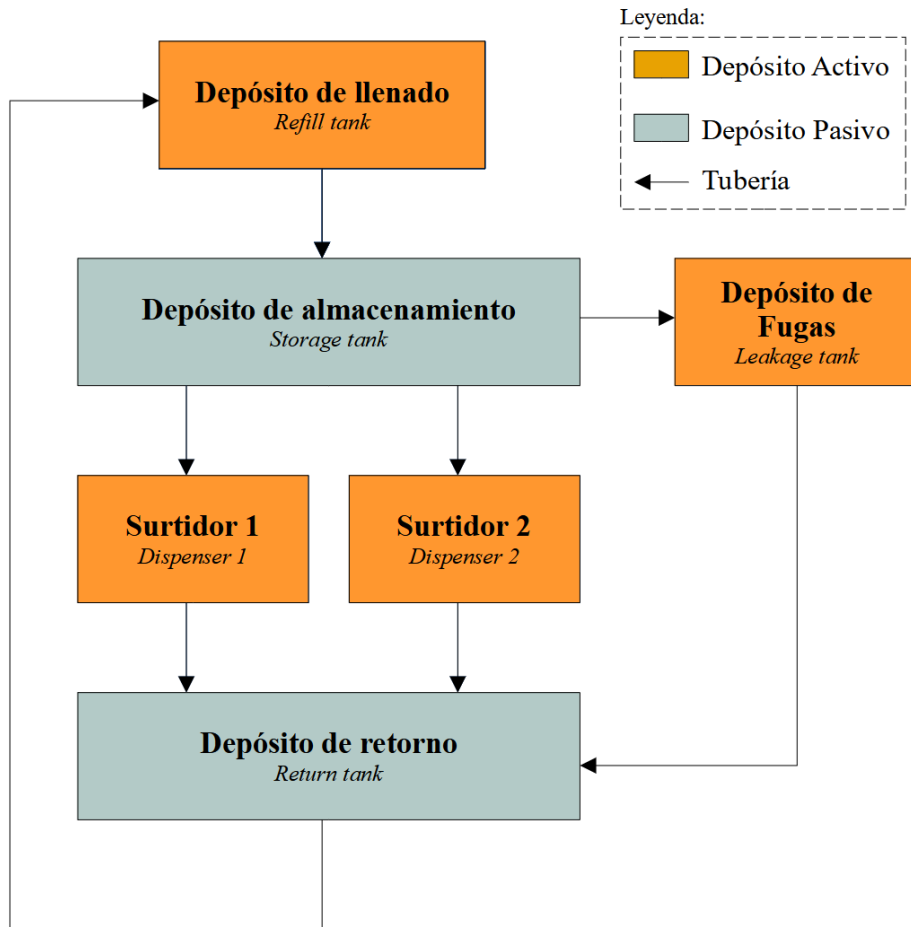


Figura 10. Esquema de la planta de la simulación

Para el caso de los depósitos activos, estos definen su comportamiento en la simulación en base a una máquina de estados de tipo Moore, es decir, por medio de una serie de estados relacionados entre sí, el depósito establece las acciones que va a llevar a cabo. El paso de un estado a otro suele definirse por una serie de eventos generados por medio de una distribución aleatoria que se explicará en el apartado 3.2.4. También interviene señales procedentes del controlador de la planta, clase que también se detalla en la sección 3.2.5. Los estados definidos son los siguientes y sus condiciones para transitar de estado:

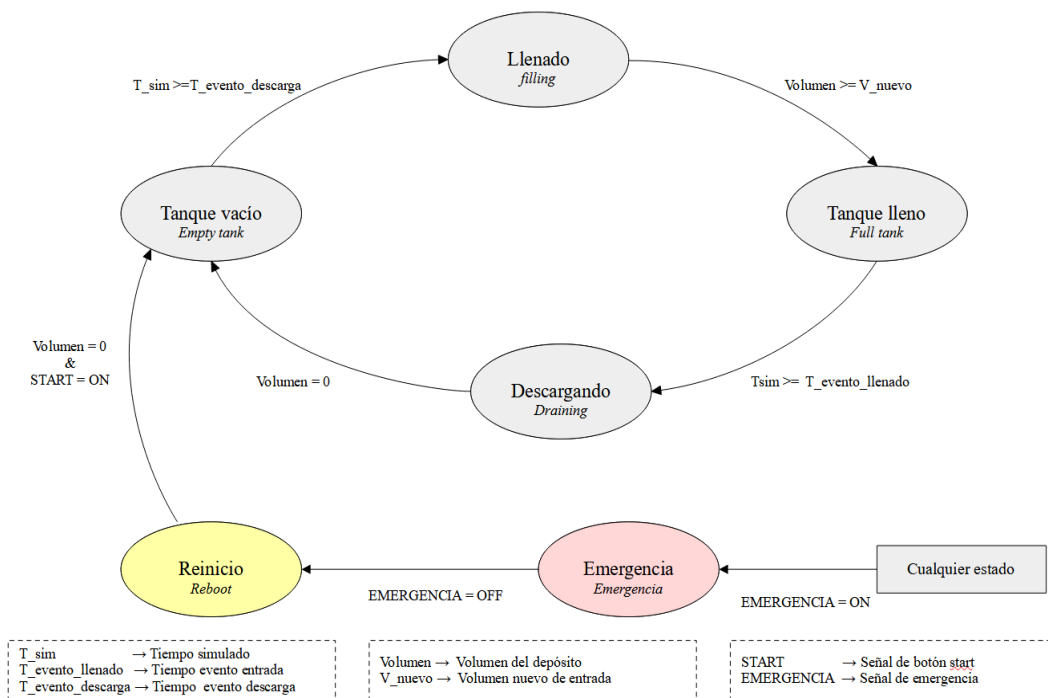


Figura 11: Máquina de estados de un depósito activo

- **TANQUE VACÍO:** Estado inicial de todos los depósitos activos. Mantiene las válvulas de llenado y vaciado cerradas hasta que sea el momento de empezar el proceso de llenado, que es determinado por un evento de tiempo de entrada de líquido en el depósito.
- **LLENADO:** En este estado se abre la válvula de llenado y se mantiene cerrada la de vaciado hasta que el depósito se llene con una cantidad de líquido definida por el evento de volumen de entrada del depósito.
- **TANQUE LLENO:** El depósito mantiene ambas válvulas cerradas hasta que la simulación diga que es el momento de descargar, definido por un tiempo generado por el evento de salida del depósito activo.
- **DESCARGANDO:** En este estado se abre la válvula de vaciado y se mantiene cerrada la de llenado hasta que el depósito quede vacío y vuelva al estado inicial.

- EMERGENCIA: Los depósitos entran en este estado de emergencia si se ha activado la seta o si han sobrepasado su volumen de seguridad. Mantiene todas las válvulas cerradas.
- REINICIO: Una vez se ha salido del estado de emergencia, se abren todas las válvulas del sistema para vaciar todos los tanques. Una vez finalizado el vaciado, se puede volver a empezar la simulación.

3.2.2. Tuberías y caudales

Como se ha indicado anteriormente, las tuberías son los objetos que tienen la función de transferencia de volumen, y controlan esta transferencia en base a una válvula ficticia. Si la válvula de la tubería está cerrada no se realiza ningún cambio del volumen y viceversa. Para definir el modo que realiza esta transferencia utilizamos otro objeto asociado a la tubería, el caudal.

El objeto caudal se puede definir por dos tipos de flujos: el caudal constante que viene definido simplemente por un valor constante R (l/min) y el flujo variable que viene determinado por la expresión:

$$Q_{variable} (L/min) = R \cdot \sqrt{\frac{Volumen}{Máx. volumen}} \quad (1)$$

Siendo R la constante del caudal (L/min), el volumen y el máximo volumen correspondientes al volumen del depósito al cual se le extrae el líquido. Este caudal variable es definido concretamente para la tubería que provoca la fuga del depósito de almacenamiento.

3.2.3. Ruido

En la simulación, la transferencia y medida del volumen ocurren de forma ideal. No existe ninguna interferencia o error en las medidas. Por ello, para obtener una muestra de datos con un comportamiento real de una planta, incluimos en la simulación el objeto de ruido. Este aportará un error en la medida de volumen del depósito de almacenamiento, en las transferencias de combustible en la venta y recarga del depósito de almacenamiento. En todos los casos se utiliza el ruido Gaussiano o también conocida como distribución normal [14].

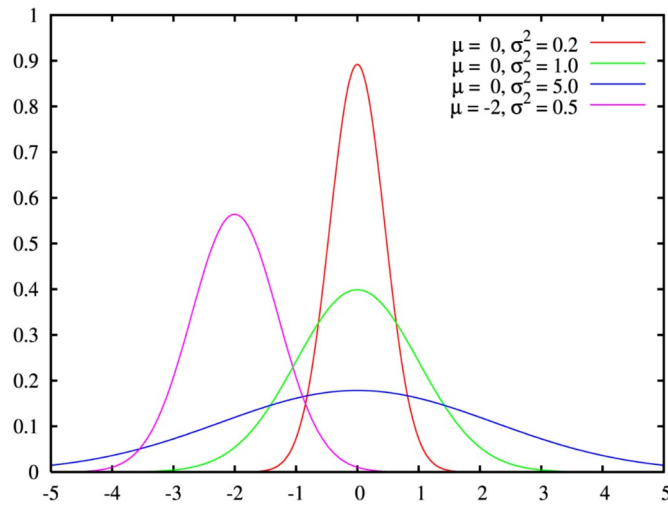


Figura 12: Distribución normal

Sus parámetros son la media y su varianza y los establece el usuario en el fichero de configuración.

3.2.4. Eventos

En la simulación utilizamos eventos de distribución aleatorias para simular las ventas en los depósitos de los surtidores y las recargas de las cubas en el depósito de almacenamiento. Por ello hay dos tipos de eventos:

- Los eventos de tiempos de las entradas y salidas de líquido de los tanques activos. Estos establece el tiempo cuando inician carga y descarga en el tiempo de simulación
- Los eventos de volumen de entrada de los depósitos activos, que determinan la cantidad de volumen recibirán

Nótese, que no hay un evento de volumen de salida en los depósitos porque en la simulación los tanques activos una vez recargados se vacían completamente en el estado descargando por lo que no es necesario un evento del volumen en la descarga. Para esta simulación, se emplea la distribución normal como distribución aleatoria para los eventos, tanto para volumen como de tiempo.

3.2.5. Controladora

La controladora es un objeto que se dedica a la gestión del funcionamiento de la planta simulada y se encargará de definir la apertura y cierre de las válvulas de la planta. También

influye en los estados de los propios depósitos. Para definir su modelo de comportamiento utilizamos una máquina de estados de tipo Moore, al igual que los tanques activos (Figura 13).

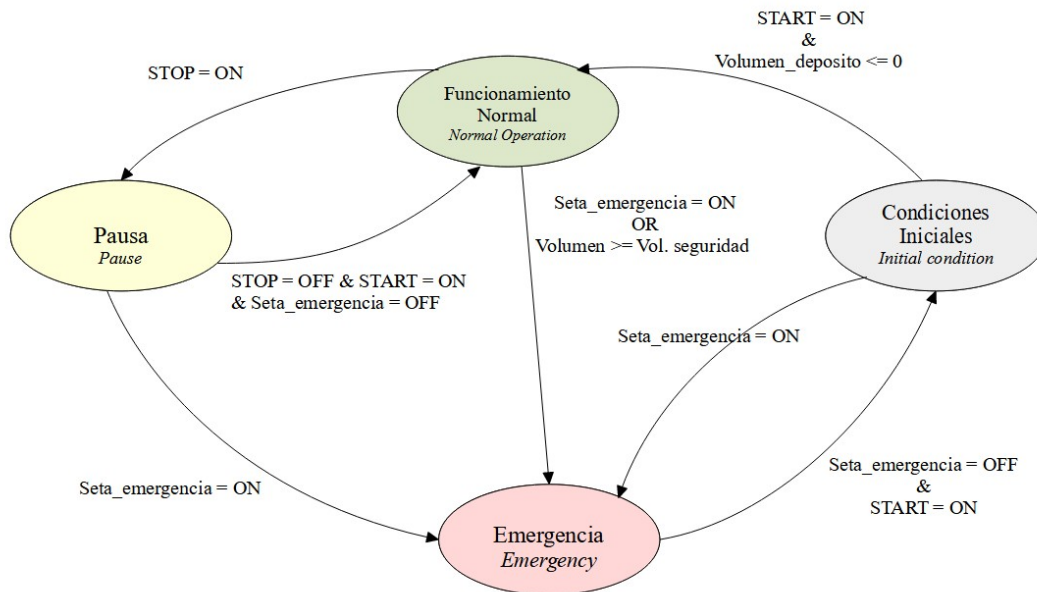


Figura 13. Máquina de estados de la controladora

El objeto controladora tiene acceso a los atributos y funciones de los depósitos de la planta y además tendrá asociado una serie de botones que determinarán su funcionamiento, simulando como un cuadro de control de la planta. Se describen cada uno de ellos:

- Botón START. En caso que el sistema esté en pausa o en condiciones iniciales reanuda a un modo funcionamiento normal.
- Botón STOP: Detiene el proceso si está operando de forma normal. Es posible reanudar el funcionamiento normal, si es detenido vía este botón.
- SETA DE EMERGENCIA: Detiene la operación de la planta desde cualquier estado y establece el estado de Emergencia. Para salir de este estado hay que, poner a 0 esta variable y activar la variable botón START.

Los estados de la máquina de la controladora son los siguientes:

- **FUNCIONAMIENTO NORMAL:** es el estado, en el que la planta opera de forma normal para simular la estación, en él, los depósitos activos actualizan sus estados para definir el comportamiento de las tuberías de sus válvulas. Desde este estado se puede entrar en el estado de PAUSA si se acciona el pulsador de STOP.
- **PAUSA:** En este estado la simulación se detiene, ya que todas las válvulas se cierran y los depósitos se mantienen esperando a que el sistema retorna al FUNCIONAMIENTO NORMAL cuando la señal de STOP esté desactivada y se active la señal de START.
- **EMERGENCIA:** se accede a este estado cuando se activa la señal de seta de emergencia o un depósito sobrepasa el volumen de seguridad. Esto provoca que todos los depósitos entren en estado de emergencia. Se mantienen en ese estado hasta que se active la señal START y se desactive la señal de parada de emergencia. Hay que añadir que se puede transitar a este estado desde cualquier otro existente.
- **LLEVAR A CONDICIONES INICIALES:** Es el estado previo para preparar el inicio del Funcionamiento Normal. En este estado se abren solo las válvulas de depósitos de salida para vaciar los tanques activos y que solo tenga líquido los depósitos pasivos.

3.2.6. Planta

La clase planta contiene toda las clases anteriores para crear la planta para la simulación y ella establece funciones para la ejecución de la simulación y funciones para almacenamiento datos dentro de la simulación con el fin de generar un exportable de datos.

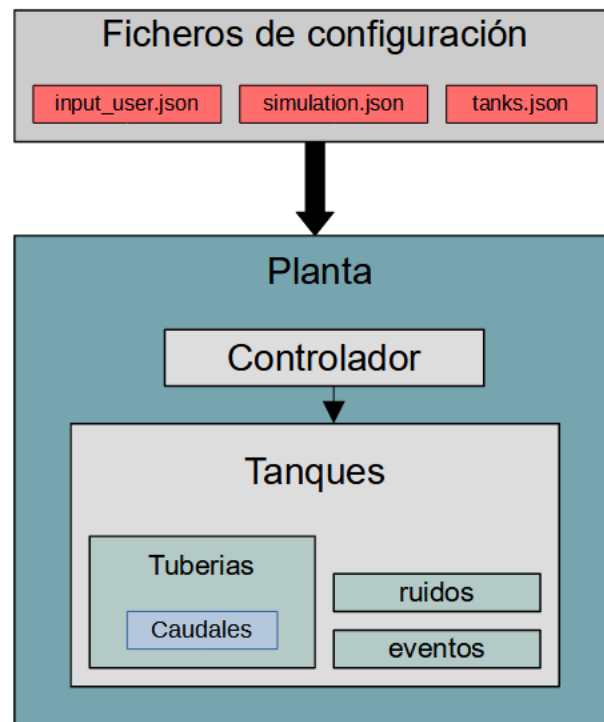


Figura 14: Esquema de la clase planta

Para la creación del objeto de esta clase se necesita los datos de configuración del usuario.

3.3. Funcionamiento del simulador de la planta

3.3.1. Ficheros de configuración del usuario

Previo a la ejecución de la simulación, el usuario puede modificar características propias de la simulación, en los ficheros JSON del simulador.

- **Simulation.json:** fichero donde se define los parámetros de tiempo de simulación. Se establece el tiempo de paso de la simulación, los tiempos de inicio y final para el registro de inventario de la simulación. Y el tiempo de intervalo entre registros del inventario. La simulación termina con el último registro del inventario.
- **Inputs_user.json:** en este fichero, se establece la acción de los botones del panel de control de la controladora. Para este proyecto se le indica la activación del botón **START** al inicio de la simulación que comience a ejecutarse operará la planta en simulación.

- Tanks.json: finalmente, en este fichero se establecen los tanques y sus tuberías conectadas entre sí, aparte de otros parámetros de los depósitos. Para este proyecto se definió el diseño de la figura 10 dentro de este fichero para la simulación de la planta.

3.3.2. Proceso de simulación

Inicialmente, el simulador carga los datos de entrada de los ficheros de configuración de usuario y genera un objeto planta con el diseño definido en el fichero de configuración “Tanks.json”. Una vez que la simulación tiene la planta creada, procede a ejecutar el proceso de simulación en base a los parámetros de tiempos definidos en el fichero “simulation.json”.

La simulación se establece por medio de un bucle que parte con una variable de tiempo de simulación 0 y en cada iteración de este bucle incrementa su valor con un tiempo de paso, que ha sido definido por el usuario en el fichero “simulation.json”. Este bucle se detendrá, cuando el tiempo de simulación alcance el tiempo del último registro de simulación, y por tanto, deteniendo el proceso de simulación. En un ciclo de simulación se establecen las siguientes fases:

1. Lectura del panel de control. En esta fase se comprueba si en ese instante de tiempo el operario acciona alguna entrada en el panel de control y en caso que sea así se actualiza la controladora teniendo en cuenta esa nueva entrada. Por ejemplo, que el usuario haya apretado el botón STOP. Estas acciones vienen definidas en el fichero de configuración “Inputs User.json”
2. Guardado de datos de planta. Se realiza un guardado de los volúmenes de los depósitos, y otros parámetros, etc. Para la obtención del inventario de datos al final de la simulación.
3. Actualización del estado de la controladora. La controladora establece su estado en base a una serie de eventos de la simulación y a las entradas del panel de control. Y realiza acciones de control en la planta a partir de su estado actual. También se actualizan los estados de los depósitos que pueden determinar si las válvulas de las tuberías se abren o se cierran según su estado.

- Actualización de la planta. Establecidos los estados de depósitos y del controlador, se aplica una actualización de los volúmenes de los depósitos que dependen de si las tuberías se encuentran abiertas o cerradas, dado el paso anterior. En caso que exista una transferencia del volumen entre depósitos, la cantidad viene definida por el tipo de caudal o en el caso que límite el volumen del depósito a llenar o vaciar.
- Nuevo tiempo de simulación . Se incrementa el tiempo de actualización sumando el tiempo de simulación actual con de paso. Terminando el ciclo del bucle.

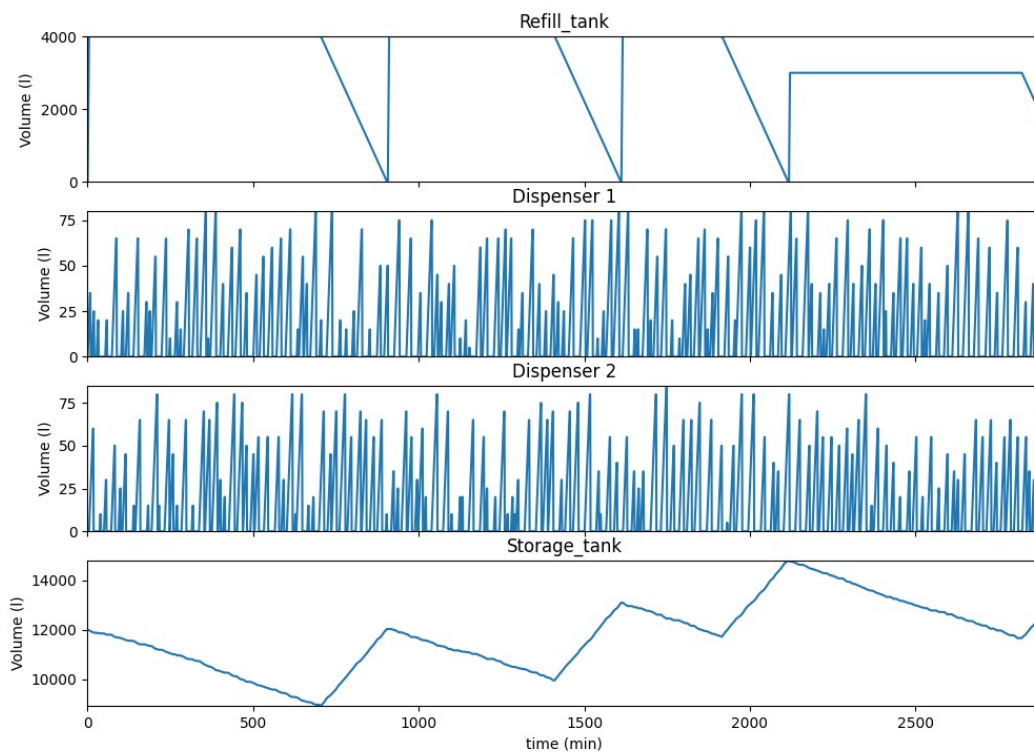


Figura 15: Representación de los datos del volumen de los depósitos del simulador

Terminada la simulación, a partir de los datos guardados se podrá obtener como resultado de la simulación una tabla de datos que simula el inventario de una estación de servicios.

3.4. Datos obtenidos

Como resultado de la simulación se crea un fichero .csv con los registros de inventario diarios de la planta, dado el tiempo definido en los ficheros de configuración de usuario. En este caso, para su posterior uso en la detección de fugas.

Tiempo (días)	Volumen depósito almacenam. inicial (L)	Venta (L)	Llenado depósito almacenam. (L)	Volumen depósito de almacenam. final teórico (L)	Volumen depósito de almacenam. final (L)	Variación (L)	Variación Acum. (L)	Fuga
...
12	22210,68	-5940	8025,72	24296,4	24259,42	-36,98	-90,99	0
13	24259,42	-6120	5930,41	24069,83	24149,95	80,12	-48,86	0
14	24149,95	-6145	3361,09	21366,04	21338,49	-27,55	-28,9	0
15	21338,49	-5875	6628,99	22092,48	22017,06	-75,42	-92,81	1
16	22017,06	-5880	6527,3	22664,36	22677,09	12,73	-64,87	1
17	22677,09	-6000	3899,85	20576,94	20632	55,06	-18,46	1
...

Tabla 1. Fragmento de tabla de datos de la simulación de la estación de servicios

Los datos que componen el registro de inventario son los siguientes:

- Tiempo (días) : tiempo medida de la simulación
- Volumen depósito almacenamiento inicial (L): volumen inicial al comienzo del día.
- Ventas (L): las ventas totales ocurridas en el día
- Descargas (L): la cantidad de litros alimentados al depósito almacenamiento por un camión cisterna en el día.
- Volumen depósito de almacenamiento final teórico (L): volumen calculado en base al volumen depósito al inicio del día, las ventas ocurridas y el llenado del depósito de almacenamiento.
- Variación (L): la resta entre la volumen depósito de almacenamiento final teórico y real.

- Variación acumulada (L) : suma de n días de las variaciones de días anteriores (ventana de tiempo que nos ayudará en la detección de las fugas).
- Fuga: variable binaria que indica si hay fuga o no en el depósito (1: se produce una fuga / 0: No se produce ninguna fuga).

Una vez obtenido los registro de inventarios, se procede a la fase del proyecto de utilizar *Machine Learning* para la detección de fugas.

4. DETECCIÓN FUGAS POR MEDIO MACHINE LEARNING

4.1. Aplicación de Machine Learning en la detección de fugas

4.1.1. Machine learning (ML)

En los últimos años, debido al incremento de datos (conocido como *big data*) y a la mejora de la capacidad de procesamiento de los equipos computacionales, se ha generado enormes avances en el campo de la Inteligencia Artificial (IA) y sobre todo, en una de sus ramas, el *Machine Learning* o aprendizaje automático.

Esta rama de la IA se basa en la construcción de sistemas que puedan aprender automáticamente de la experiencia y datos recibidos. Esto implica, identificar patrones complejos y por medio de algoritmos que predicen comportamientos, mejora sus procesos automatizados. Estos algoritmos se clasifican en tres categorías en función de su modo de aprendizaje con los datos que se utilizan [15]:

- Supervisado. Este tipo de aprendizaje se emplea entrenando un modelo con conjuntos de datos en los que se conoce el valor objetivo que se desea reproducir. Pudiéndose medir su desempeño en base a lo bien que reproduce este valor objetivo.
- No supervisado. Se busca, en este caso, que los modelos reproducen un valor concreto. Siendo el objetivo de estos modelos identificar patrones que permitan separar y clasificar los datos en diferentes grupos.
- Por refuerzo. Se busca la creación de agentes que pueden realizar acciones sobre un entorno. Siendo este un aprendizaje completamente diferente a los dos anteriores.

Para el caso del problema de detección de fugas, el tipo de aprendizaje que se utilizará en nuestro proyecto será el supervisado, ya que disponemos de los datos y del valor objetivo que queremos reproducir (las fugas).

Centrándonos en los modelos de aprendizaje supervisado, estos se pueden clasificar en dos tipos según el dato que se desea predecir [17]:

- **Regresión:** modelos que predicen valores reales continuos y no discretos. A partir de los datos de entrenamiento el modelo construye las características del algoritmo y los valores de salida. Posteriormente se le insertan nuevos datos para comprobar las predicciones del modelo entrenado.
- **Clasificación:** son modelos que predicen etiquetas, siendo posible determinar dos clases lo que implicaría que es un modelo de clasificación binaria, o más clases lo que significa que sería multiclase. Al igual que el caso anterior, el modelo aprende por medio de datos de entrenamiento con la salida que debería devolver y luego se comprueba su eficacia con datos nuevos.

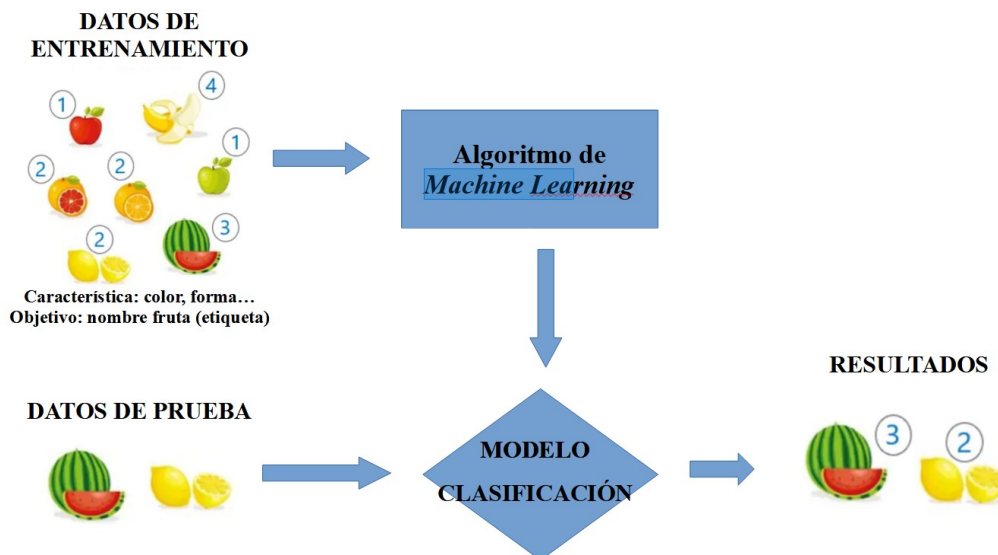


Figura 16. Esquema de entrenamiento de un modelo de clasificación multiclase

Como en este caso debemos de identificar si hay o no una fuga, siendo 0 no hay y 1 en caso contrario, se emplearán modelos de clasificación binaria para el desarrollo del sistema de detección de fugas.

4.1.2. Aplicaciones previas de ML en la detección de fugas

Como se ha mencionado en anteriores apartados de la memoria, este proyecto parte de una serie de estudios sobre la aplicación de *Machine learning* para la detección de fugas de combustible de estaciones de servicio, en los que los autores usan clasificadores binarios para determinar por medio del entrenamiento con datos de libros de inventario de una estación, si existe una fuga o no [5][6].

Los resultados obtenidos fueron bastante positivos y descubrieron que con la aplicación de la aplicación de una “ventana de tiempo” en un campo calculado, se consigue detectar fugas pequeñas sin sobrepasar el tiempo límite por los marcos europeos que regulan el tiempo máximo de detección de fuga en los depósitos, que es de máximo 14 días con una fuga de 19,2L/día (UNE-EN 13160-5, 2005) [1]. El susodicho campo se calcula por medio de la diferencia del volumen del depósito de almacenamiento medido y el volumen calculado (por medio del volumen inicial del día, las ventas y recargas del día) y se le aplica la ventana de tiempo que implica sumar un acumulado de está diferencia de n días anteriores, obteniendo así una nueva característica que ayudará al modelo de clasificación detectar la detección de fugas.

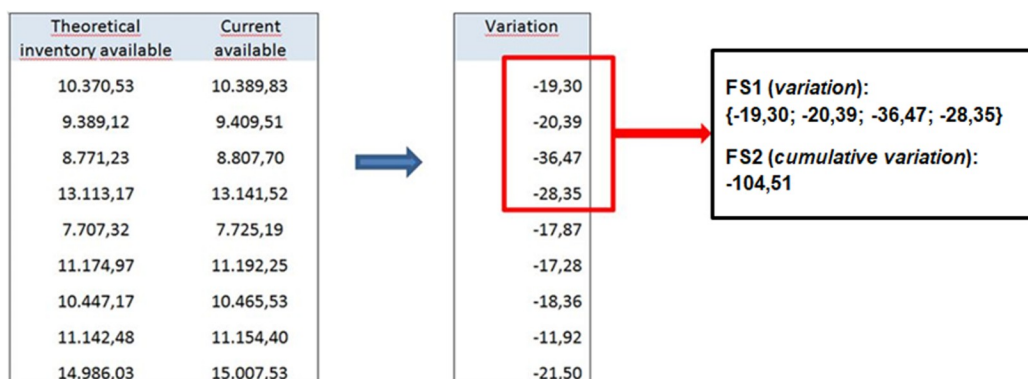


Figura 17. Ejemplo de la ventana de tiempo.

Para nuestro sistema de detección de fugas, con los datos inventario de la simulación se calculará la variación de volumen medido con respecto el calculo del depósito y aplicaremos una ventana de tiempo de 10 días con el objetivo de tener un buen resultado en la detección de fugas por medio de nuestro sistema.

4.2. Desarrollo del sistema de detección fugas

4.2.1. Diseño del sistema ML de detección de fugas

El sistema de *Machine Learning* se basará en la aplicación del siguiente procedimiento:

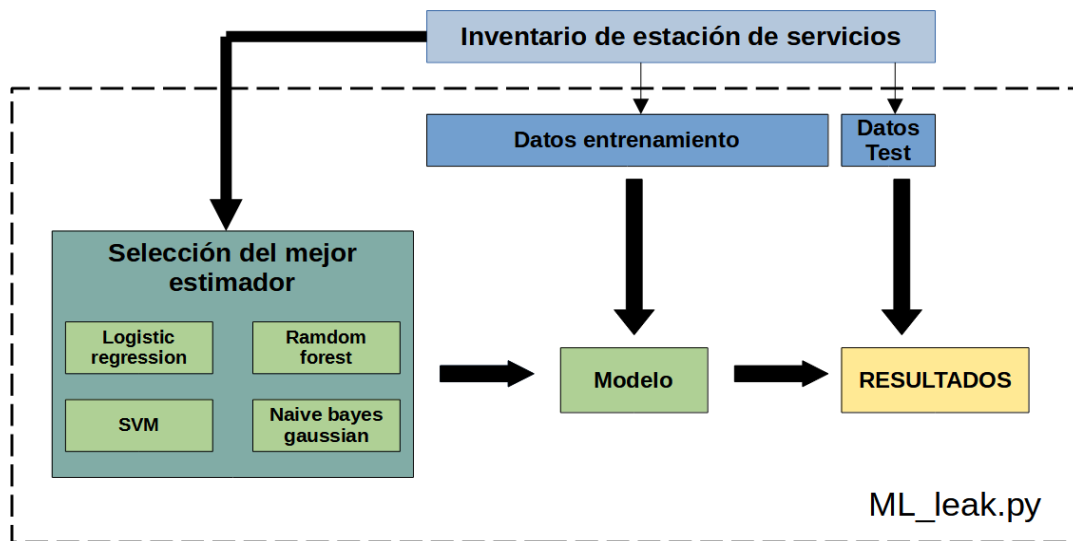


Figura 18. Esquema del sistema de detección de fugas aplicando ML

1. Al sistema se le introduce un inventario de datos de una estación de servicios del que se desea obtener un modelo de detección de fugas basados en esos datos.
2. Para determinar el mejor estimador para crear el modelo, se utiliza una función que nos permite realizar evaluaciones con distintos estimadores cambiando parámetros configurados. Como resultado de esta función podemos obtener cual es el mejor estimador y sus parámetros más óptimos usando los datos de inventarios introducidos en el punto 1.

3. Conociendo el mejor estimador, sea crea un modelo de clasificación en base a este y lo entrenamos con un segmento de datos del inventario que nos interese detectar fugas. Una vez hecho esto se comprueba un rendimiento con una selección de datos de prueba para comprobar la eficacia de la estimación realizada.

4.2.2. Modelos de clasificación

Los tipos de estimadores que usan en nuestro sistema para generar el modelos de clasificación son los siguientes:

- **Regresión logística (logistic regression)**. Es un método de regresión que permite estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa. [20]
- **Bosques aleatorios (random forest)**. Es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de *bagging* que construye una larga colección de árboles no correlacionados y luego los promedia. [21]
- **Naive bayes gaussian**. Es un clasificador probabilístico simple basado en la aplicación del teorema de Bayes con suposiciones fuertes (ingenuas) de independencia entre las características. El uso del teorema de Bayes en la regla de decisión del clasificador determina la probabilidad de que un objeto pertenezca a una clase, basándose en el conocimiento previo de las condiciones que podrían estar relacionadas con la clase [22].
- **SVM (Support vector model)**. SVM considera cada objeto de datos como un punto en el espacio n-dimensional (donde n es el número de funciones), con el valor de cada función siendo el valor de una determinada coordenada. La clasificación es entonces llevado a cabo encontrando el hiperplano que mejor separa los dos clases en el espacio [23].

4.2.3. Medición del rendimiento de los clasificadores

Para poder medir el rendimiento del modelo se debe establecer un sistema de medición previamente. Para los clasificadores binarios las métricas se calculan a partiendo de la matriz de confusión [18]:

	1	0
1	True positive (TP)	False positive (FP)
0	False Negative (FN)	True Negative (TN)

Tabla 2. Matriz de confusión

Siendo los números de la primera columna, los valores predichos y los valores de la primera fila, los valores reales. Esta tabla contempla los cuatro casos posibles en una predicción del modelo. Y partiendo de esta matriz, las métricas del clasificador que se definen son las siguientes:

- *Accuracy*: Representa el porcentaje total de valores correctamente clasificados, tanto positivos como negativos. Esta métrica es recomendable si en los datos la cantidad de cada etiqueta esta bien balanceada.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (2)$$

- *Precision Score*. Permite conocer el porcentaje de valores que se han clasificado como positivos sean realmente positivos.

$$Precision = TP / (TP + FP) \quad (3)$$

- *Recall Score*: Ratio de verdaderos positivos.

$$Recall = TP / (TP + FN) \quad (4)$$

- *F1 Score*. Esta es una métrica muy utilizada en problemas en los que el conjunto de datos a analizar no tienen una cantidad de etiquetas equilibradas.

$$F1 = 2 * ((recall * precision) / (recall + precision)) \quad (5)$$

Estas métricas son útiles si te interesa específicamente la evaluación del modelo que aportan, o porque ya conoces la distribución de las etiquetas que debería predecir el modelo y podemos seleccionar un método de puntuación óptimo.

No obstante el mejor índice para medir la eficacia del modelo de forma genérica es aplicando la métrica AUC-ROC (Área bajo la curva de característica de funcionamiento del receptor) y es el que utilizamos para la estimación del mejor modelo en nuestro programa.

Como indica su nombre esta métrica se calcula con el área bajo la curva ROC. Una curva ROC (*Receiver Operating Characteristic*) es un gráfico utilizado para evaluar modelos de Machine Learning para problemas de clasificación. La gráfica representa el porcentaje de verdaderos positivos, *Recall*, frente al ratio de falsos positivos (*False Positive Rate*). La diferencia con el resto de métricas, es que en este caso, el umbral por el que se clasifica un elemento como 0 o 1, se va modificando, para poder ir generando todos los puntos de la gráfica.

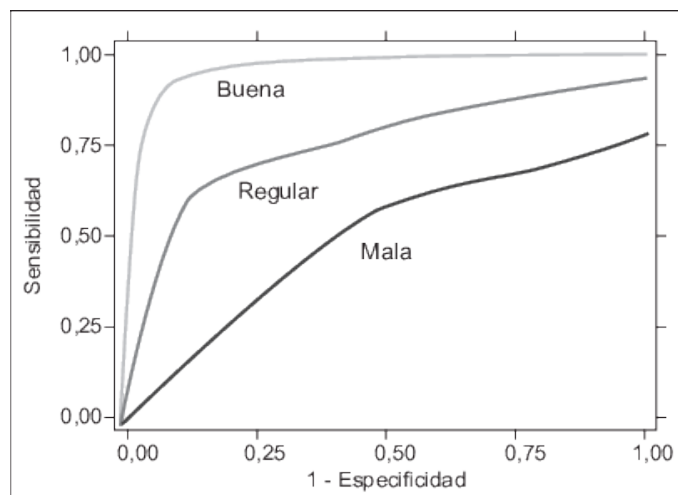


Figura 19. Gráfica Curva ROC

Afortunadamente, en la librería *scikit-learn* de python para aplicar ML se dispone de una función que te devuelve la puntuación en base a esta métrica (*roc_auc_score*)

4.2.4. Método de selección del mejor estimador

El procedimiento de selección del mejor estimador se basa realizando entrenamientos de modelos del mismo estimador pero con distintos parámetros de configuración y comparando estos modelos en función de una métrica de rendimiento (*roc_auc_score*), se identifica los parámetros de configuración óptimos de ese estimador. Esto se aplica para los demás estimadores y finalmente se comparan de cada estimador el que mejor resultado a obtenido y el que posea el mejor resultado es el seleccionado como mejor estimador para crear el modelo con el que se entrenará.

Durante el proceso de evaluación de los modelos, el entrenamiento de un modelo con los datos del inventarios requiere de segmentar un subconjunto para evaluar el rendimiento del modelo, pero este método tiene un problema, y es que los datos entrenados no necesariamente tienen la información para que modelo aprenda del subconjunto que se le ha extraído, dando un posible equívoco mal resultado de evaluación de la puntuación. Por otro lado, no es correcto entrenar con todos los datos a la vez porque tienes un modelo validado en base a datos que ya conoce. Para solventar este problema se le aplica la técnica de validación cruzada k-fold (*K-fold cross validation*)

Esta técnica realiza un número k entrenamientos con distintos modelos del mismo estimador, variando en cada iteración el conjunto de datos de validación. Con todos los resultados hacemos una media de las distintas puntuaciones del modelo. Este método consigue una puntuación más estable y fiable del modelo con los datos aportados.

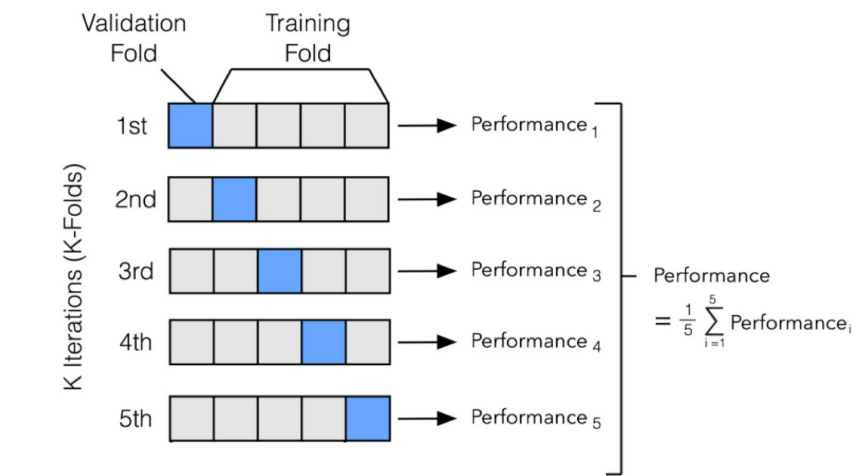


Figura 20. Validación cruzada k-folds (K=5)

4.3. Resultados obtenidos

Por medio de la simulación hemos obtenido dos inventarios para probar dos situaciones distintas y verificar la capacidad del sistema desarrollado para detectar fugas. En ambos casos, la simulación de la fuga se le aplica un ratio de fuga nominal del 19,2 L/día, en base a normativa UNE-EN 13160-5, 2017 [1] y una ventana de tiempo de 10 días en la característica de variación del volumen mencionado en el apartado 4.1.2.

4.3.1. Pruebas utilizando inventario anual

En este caso el inventario tiene un tiempo total de registro de un año, tomando un registro por día, es decir, 365 registros. Con esta prueba veremos como se comporta los modelos con amplio margen de tiempo. Las fugas que se simulan a lo largo del año no es continua sino que hay períodos que se detiene la fuga para simular que la fuga se detiene debido a factores externos como cambios presión en el depósito, etc. El resultado obtenido aplicando el método fue el siguiente.

```

                MODELO  MEJOR PUNTUACIÓN
0                svm      0.988556
1      random_forest    0.984388
2  naive_bayes_gaussian 0.991468
3  logistic_regression  0.992147

EL MEJOR ESTIMADOR ES : logistic_regression

PUNTUACIÓN DEL MODELO CON MUESTRAS DE PRUEBA:
scoring roc auc : 0.96
Accuracy Score : 0.959
Precision Score : 0.978
Recoll Score :0.957
F1 Score : 0.968
```

Figura 21. Resultados del modelo con un inventario de un año

Vemos que el sistema detecta que el mejor estimador es el de regresión logística y testeando el modelo con un subconjunto con el que no entrena vemos que la detección de fugas lo realiza con márgenes de error menor del 5%.

4.3.2. Pruebas utilizando inventario mensual

En esta prueba hemos creado un inventario de 28 días de duración, cuyo intervalo de registro es diario. En esta simulación las fugas se simulan a partir del día 14 y dura hasta terminar el inventario para comprobar si el modelo entrenado es capaz de identificar una fuga en base a ese rango de tiempo. Esto se aplica ya que la normativa indica que debe ser detectado no más tarde de 14 días desde que sucede la fuga [1]. El resultado obtenido es el siguiente:

```

                MODELO  MEJOR PUNTUACIÓN
0                svm          0.8
1      random_forest        1.0
2  naive_bayes_gaussian    0.9
3  logistic_regression      0.9

EL MEJOR ESTIMADOR ES : random_forest

PUNTUACIÓN DEL MODELO CON MUESTRAS DE PRUEBA:
scoring roc auc : 1.0
Accuracy Score : 1.0
Precision Score : 1.0
Recoll Score :1.0
F1 Score : 1.0
```

Figura 22: Resultados del modelo con un inventario mensual

A pesar del ruido añadido tanto en la recarga del depósito de almacenamiento, en la medida de su propio volumen o en las pérdidas en la dispensadores, el sistema es capaz de detectar perfectamente cuando se produce una fuga.

4.3.3. Análisis de los resultados

Ambos casos han dado muy buenos resultados, no obstante, es posible que se deba por la implementación de los ruidos de la simulación. Ya que éstos se modelan por medio de distribuciones normales y por ello, los modelos clasificación pueden aprender más fácilmente comparado con datos reales que pueden tener errores humanos en los registros o fallos aleatorios en los equipos que no siguen distribuciones estadísticas y dificultan más el aprendizaje.

5. CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

5.1. Conclusiones

Una de los mayores desafíos en la realización de este del proyecto ha sido el desarrollo del simulador de la planta. La conversión del diseño del prototipo de planta de una estación de servicios basado en el trabajo final de grado “*Diseño e implementación de un sistema autónomo para la simulación de fugas en depósitos*” [10], requirió de varias revisiones y cambios dentro del diseño de nuestra simulación. Esto causó que, gran parte del tiempo dedicado al desarrollo del proyecto se centrará en la elaboración de este simulador.

Como resultados, se ha obtenido un sistema que nos permite reproducir datos en una tabla de inventario similar a la de una estación de servicios real y además con la ventaja de tener una mayor capacidad de parámetros configuración. Y también, hemos desarrollado un sistema que nos permite identificar cual es el mejor estimador para la detección de fugas y sus mejores parámetros en base al inventario de datos simulados que se le introduce. Por tanto, se han completado los objetivos con los que se pretendía realizar con el proyecto. Sin embargo, el proyecto podría haber sido más ambicioso y haber abarcado más en el ámbito de *Machine Learning* aplicar nuevas metodologías de identificación de fugas y realizar comparaciones con las aplicadas en este proyecto, lamentablemente, no se pudo desarrollar por falta de tiempo.

Concluido este proyecto, personalmente opino que la aplicación de *Machine Learning* en sectores industriales, tendrán un fuerte impacto en los próximos años gracias a las nuevas herramientas que a día de hoy ya se disponen y sin requerir elevados costes económicos. Por ello, es atractiva la investigación del *Machine Learning* pues potencialmente se puede desarrollar modelos y herramientas de predicción que puede ahorrar costes de mantenimiento de equipos, optimizar procesos, o incluso evitar desastres ambientales como este caso.

5.2. Conclusions

One of the biggest challenges in carrying out this project has been the development of the fuel station simulator. The conversion of the simulation design implemented based on the final degree project "Design and implementation of an autonomous system for the simulation of leaks in tanks" in a single programming language, involved several revisions and changes within the simulation. This meant spending a lot of time on this part of the project.

As a result, a system has been obtained that allows us to reproduce data in an inventory table similar to a real petrol station and it has advantages such as a greater capacity for configuration parameters. Also, we have developed a system that allows us to identify the best parameters and the best classifier model for leak detection based on the simulated storage data inventory. However, the project could have been more ambitious and it could have covered more in the knowledge of Machine Learning, such as new methodologies for identifying leak patterns, which could not be developed due to lack of development time.

Personally, I think that the application of Machine Learning in industrial activities will have a great impact in the next few years, thanks to new tools that we can use. For this reason, it is interesting to carry out new research on the subject, since it will allow the development of new tools that will save on maintenance costs, improve resource optimization or even avoid environmental disasters such as this applied case.

5.3. Mejoras y nuevas líneas de trabajo

En base al proyecto desarrollado, se puede plantear nuevos proyectos que podrían ampliar las posibilidades del trabajo realizado. A continuación, se expone las siguientes opciones:

5.3.1. Actualizar, mejorar, completar el simulador

Se pueden añadir varias mejoras .

- Añadir más tipos de ruido, de caudal, o incluso de eventos de volumen para los depósitos, lo que permitiría tener más opciones a configurar dentro de la simulación.
- Desarrollar una interfaz gráfica para que al usuario le sea más amigable utilizar la simulación.
- Ofrecer más opciones en la opción de exportar datos, no solo los libros de inventarios de una estación de servicios si se requiriera.
- Mejorar ciertas limitaciones de la simulación. Por ejemplo las válvulas en el simulador solo tienen dos estados y no existe una transición intermedia de estos dos estados (abierto o cerrado). Otro mejora sería contemplar la presión de los depósitos para la dinámica en el proceso.

5.3.2. Aplicar nuevas técnicas de ML

Ante el incremento del desarrollo de esta rama de la inteligencia artificial, en un futuro sería interesante hacer un nuevo proyecto aplicando las nuevas herramientas o conocimientos que hayan podido surgir entonces, para volver aplicarlas al caso de detección de fugas. O estudiar otras alternativas actuales a las implementadas en este trabajo.

5.3.3. Realizar otros estudios en relación a la detección de fugas

Una de las ventajas del simulador es que podemos contemplar múltiples escenarios para hacer pruebas en la detección de métodos de fuga, y gracias a esto, podemos ver que efecto tiene ciertos parámetros para la propia detección. Por ejemplo se puede realizar un estudio en base a los distintos tipos de ruidos aplicados en las medidas y analizar el impacto tiene en la detección según el estimador utilizado y el ruido definido.

ANEXOS

Anexo 1. Presupuesto

Una de las ventajas de este proyecto es que no se requiere del montaje de un prototipo real de una planta ahorrando costes en la compra de materiales. Aunque como mínimo se necesitaba un equipo informático adecuado para desarrollar el proyecto. Este gasto puede omitirse, ya que se disponía del equipo informático antes del inicio del mismo.

No obstante, detallamos un presupuesto de un equipo informático en caso que se requiriera de uno.

Componentes	Modelo	und	€/und	Precio (€)
Procesador	Intel Core i5-11400F 2.6 GHz	1	165	165
Placa Base	Gigabyte B560M DS3H V2	1	86,99	86,99
Memoria RAM	Kingston FURY Beast DDR4 3200 MHz 16GB 2x8GB CL16	1	72,99	72,99
Torre	Tempest Spectra RGB Torre ATX Negra	1	40,99	40,99
Ventilador CPU	Tempest Cooler 4Pipes White RGB Ventilador CPU 120mm Blanco	1	30	30
Disco duro	Kioxia EXCERIA 480GB SSD SATA	1	50,99	50,99
Fuente de alimentación	Corsair CV Series CV650 650W 80 Plus Bronze V2	1	69,90	69,90
Tarjeta Grafica	MSI GeForce GTX 1050Ti 4GT Low Profile 4GB GDDR5	1	194,89	194,89
Monitor	MSI PRO MP241X 23.8" LED FullHD 75Hz	1	118,99	118,99
Teclado y ratón	Logitech MK270 Combo Teclado y Ratón Inalámbrico	1	18,98	18,98
Montaje	Servicio de PCComponentes	1	44,99	44,99
			PRECIO TOTAL	894,71

Tabla 3. Presupuesto de un equipo informático para el desarrollo del proyecto

También se expone el coste de mano de obra en base a las horas trabajadas en el desarrollo de este proyecto.

Tiempo trabajo (horas)	€/hora	Precio total (€)
300	16	4800

Tabla 4. Presupuesto correspondiente al trabajo de desarrollo del proyecto.

Como el software utilizado no requiere de ningún tipo de compra o suscripción no se detalla ningún coste al respecto.

Anexo 2. Guía del desarrollador

Anexo 2.1. Simulador de la planta

Se procede a detallar los ficheros de código que componen el simulador de una planta de una estación de servicios.

- **tank.py**: en este fichero se detallan los tipos de depósitos, sus atributos y funciones. Los atributos, independientes del tipo de depósito que sea, son su nombre, su capacidad máxima y de seguridad, el volumen actual real (no una medida), el volumen inicial del comienzo de la simulación y el controlador que está conectado.

Se definen dos tipos de depósitos:

- Depósitos pasivos (`passive_tank`): son depósitos que no tienen control en la simulación del volumen de entrada y salida del líquido almacenado.
- Depósitos activos (`active_tank`): al contrario que los pasivos, son aquellos depósitos que controlan la entrada y salida del líquido que almacenan. Por ello controlan las válvulas de entrada y salida que tienen asociadas.

Para simplificar el desarrollo del simulador, se ha establecido que las válvulas de entrada se abren o cierran todas a la vez (e igualmente para la válvulas de salida).

Tiene definido una máquina de estados que en base a su estado, establece una serie de acciones de apertura o cierre de válvula o de generación de eventos tiempo o volumen aleatorios de la simulación. Específicamente, tiene como atributos sus válvulas de entrada y salida, el estado de la máquina de estado, su estado anterior y parámetros en relación a generación de eventos.

Para transferir los volúmenes entre los depósitos se creó la clase tubería definida en el siguiente fichero.

- **pipe.py**: la clase tubería tiene como atributos un nombre asociado, el tanque que extrae su volumen, el tanque que alimenta con el volumen de anterior depósito, el tipo caudal (clase) que tiene la tubería, su estado de apertura (abierto/cerrado) y la cantidad de volumen transferido anteriormente. Dispone de una función para realizar la transferencia de volumen de los tanques conectados. La transferencia de volumen

solo puede ir en un sentido y la tubería solo se puede conectar un depósito activo con uno pasivo.

- **flow.py**: en este fichero definimos la clase caudal que nos permite crear distintos caudales para las tuberías y así definir cómo se comporta su transferencia de volumen. Sus atributos son el tipo de volumen, un valor asociado al tipo de volumen, el tanque al que se le extrae el volumen y el tiempo de simulación del tanque.

En la planta se necesita de un controlador que gestione las medidas de los elementos de la planta y aplique un control sobre la misma. Por ello, se define la clase controladora.

- **controller.py**: La clase controladora tiene una serie de atributos como el tiempo de ciclo de la simulación, los depósitos conectados, los botones del cuadro de control y el estado que se encuentra en la simulación. Tiene asociado una máquina de estado que establece el control de la planta y su funcionamiento general.

En la simulación debemos de tener en cuenta los distintos tipos ruidos que deben de interferir en las medidas de la planta por ello creamos la clase ruido en el siguiente fichero:

- **noise.py**: la clase ruido tiene como atributos el tipo de ruido y los parámetros que lo definen. Esta clase tiene una función que introduciendo un valor real como el volumen, este devuelve ese valor con un ruido aplicado que viene configurado en sus atributos.

Para la creación de la planta completa disponemos del siguiente fichero donde se define la clase planta.

- **plant.py**: fichero que define la clase planta y que dispone de la función que realiza la simulación de la misma. También dispone de funciones de guardado de datos de la simulación
- **sim_plant.py (Fichero de ejecución de la simulación)** : Engloba todos los ficheros para la simulación y genera una gráfica de la evolución del volumen de los tanques de la simulación. También exporta datos de las medidas tomadas para realizar pruebas de detección de fugas.

Anexo 2.2. Detector de fugas

Para las pruebas de detección de fugas utilizamos el siguiente script:

- **ML_leak.py** : En este script, leyendo los datos de la simulación obtenidos formato CSV, se busca identificar cual es el mejor modelo ML a implementar en la detección fugas, siendo, este un problema de clasificación binario. Para ello, se utiliza la función de la librería de scikit-learn, GridSearchCV, para evaluar y seleccionar de forma sistemática los parámetros de una serie de modelos que definimos. Y obteniendo la mejor configuración de cada modelo, selecciona el modelo con el mejor resultado para hacer una prueba de testeo con datos de prueba.

Anexo 3. Guía del usuario

Anexo 3.1. Ejecución del simulador

Previamente es necesario disponer en el equipo un entorno de python (ha sido probado con la versión 3.10) y se requiere tener instalado los siguientes paquetes instalados:

- pandas 1.4.3.
- matplotlib 3.5.3
- numpy 1.23.2

Para ejecutar el simulador solo se requiere ejecutar con el susodicho entorno de python el fichero `sim_plant.py`. Al finalizar la ejecución se obtiene un fichero de datos del inventario de la planta simulada.

El usuario puede cambiar parámetros de la configuración a través de estos tres ficheros en formato JSON:

- **simulation.json** : En este fichero se define varios parámetros de tiempo en la simulación:
 - `sim_step` (número entero) : valor que define tiempo (en minutos) del paso de la simulación
 - `log_step` (número entero) : valor de tiempo (en minutos) del intervalo en la medición de datos a obtener como resultado de la simulación.
 - `log_initial` (número entero) : tiempo inicial (en minutos) donde comienza a tomar datos de medida.
 - `log_end` (número entero) : tiempo final (en minutos) donde termina de tomar datos de medida.
- **inputs_user.json** : en este fichero se detallan las acciones que se realizan en el cuadro de mando del controlador durante la simulación. Por ejemplo, cuando se activa el botón START y se desactiva.
- **tanks.json** : Se define el diseño de la planta y sus conexiones, es decir, los tanques que lo componen y cómo están conectados entre sí. Si a los depósitos no se les define

los atributos de entrada o salida , el simulador los considerará pasivos y en caso contrario activos. En la simulación, las tuberías deben tener siempre un tanque activo y otro pasivo, no se deben conectar dos tanques activos o dos pasivos.

Anexo 3.2. Ejecución del script *Machine Learning*

Previamente es necesario disponer en el equipo informático un entorno de python (ha sido probado con la versión 3.10) y se requiere tener instalados los siguientes paquetes:

- pandas 1.4.3.
- scikit-learn 1.1.2

Para ejecutar el sistema de ML se requiere ejecutar con el entorno de python el fichero ML_leak.py.

REFERENCIAS

Documentación

[1] UNE-EN 13160-5, Sistemas de detección de fugas. Parte 5: Requisitos y métodos de ensayo/evaluación de los sistemas de detección de fugas en tanques con indicador de nivel y en los sistemas de tuberías a presión (marzo 2017).

<<https://www.en.aenor.com/normas-y-libros/buscador-de-normas/une?c=N0058060>>

[2] Real Decreto 706/2017, de 7 de julio, por el que se aprueba la instrucción técnica complementaria MI-IP 04 "Instalaciones para suministro a vehículos" y se regulan Diseño e Implementación de un sistema autónomo para la simulación de fugas en depósitos 6 determinados aspectos de la reglamentación de instalaciones petrolíferas.

<https://www.boe.es/diario_boe/txt.php?id=BOE-A-2017-9188>

[3] Real Decreto 2085/1994, de 20 de octubre, por el que se aprueba el Reglamento de Instalaciones Petrolíferas.

<<https://www.boe.es/buscar/doc.php?id=BOE-A-1995-2122>>

[4] Reglamento y normas de presentación de los trabajos fin de máster. Máster en ingeniería industrial. Escuela de doctorado y estudios de postgrado. Universidad de la Laguna. (17 de Julio de 2019).

<[https://riull.ull.es/xmlui/bitstream/handle/915/17198/Reglamento%20de%20Trabajo%20Fin%20de%20M%C3%A1ster%20\(M%C3%A1ster%20en%20Ingenier%C3%ADa%20Industrial\)%20\(2\).pdf](https://riull.ull.es/xmlui/bitstream/handle/915/17198/Reglamento%20de%20Trabajo%20Fin%20de%20M%C3%A1ster%20(M%C3%A1ster%20en%20Ingenier%C3%ADa%20Industrial)%20(2).pdf)>

[5] Alayón, S., Sigut, M., Arnay, R., Toledo, P., 2020. *Time windows: The key to improving the early detection of fuel leaks in petrol stations*. Safety Science, pp

[6] Sigut, M. Alayón, S., Hernández, E., 2014. *Applying pattern classification techniques to the early detection of fuel leaks in petrol stations*. J. Clean. Prod. 80, 262-270.

[7] Europapress, Madrid, (2016, Diciembre 1), “*Piden 4 años para el administrador de una gasolinera acusado de contaminar el Jarama*”.

<<https://www.europapress.es/nacional/noticia-piden-anos-administrador-gasolinera-acusado-contaminar-jarama-20161201174949.html>>

[8] Moya Turbica, A., Escola Tècnica Superior d’Enginyeria Industrial de Barcelona, (2003, Octubre) “*Ampliación de las instalaciones de un supermercado: Construcción de una gasolinera*”, pp 6-7, 18-19.

[9] Sánchez Mejía E.J, Luis Marín H.A., Bogotá, Universidad Católica de Colombia, 2017, “*Análisis de la amenaza presente en la infiltración de combustibles líquidos en las estaciones de servicio de la ciudad de Barranquilla*”, pp 24-26.

[10] Arriaga Campos, Luis. Universidad de La Laguna, 2020. “TFG. Diseño e implementación de un sistema autónomo para la simulación de fugas en depósitos”.

[11] Europapress, Andalucía, (2022, Agosto 23). “*Los afectados por la contaminación de Coria (Sevilla) piden prorrogar la instrucción del caso*”.

<<https://www.europapress.es/andalucia/sevilla-00357/noticia-afectados-contaminacion-coria-sevilla-piden-prorrogar-instruccion-caso-20220823150614.html>>

[12] de TechTarget, C. (2021, mayo 17). *Programación orientada a objetos, OOP*. ComputerWeekly.es.

<<https://www.computerweekly.com/es/definicion/Programacion-orientada-a-objetos-OOP>>

[13] Gustavo Pensa (2020, agosto 17). *Simulación por eventos discretos y cuándo simular*. Atlas Consultora.

<<https://www.atlasconsultora.com/simulacion-por-eventos-discretos-y-cuando-simular/>>

[14] Wikipedia contributors. (2022, Septiembre). *Distribución normal*. Wikipedia, The Free Encyclopedia.

<https://es.wikipedia.org/wiki/Distribuci%C3%B3n_normal>

[15] Rodríguez, D. (Septiembre 2022) *Machine learning*. Analytics Lane; .

<<https://www.analyticslane.com/tag/machine-learning/>>

[16] I. *Supervised learning*. (5 Septiembre 2022). Scikit-Learn.

<https://scikit-learn.org/stable/supervised_learning.html>

[17] Gonzalez, L. (2018, agosto 3). *Todo sobre aprendizaje supervisado en Machine Learning - Aprende IA*. Aprende IA.

<<https://aprendeia.com/todo-sobre-aprendizaje-supervisado-en-machine-learning/>>

[18] Díaz, R. (2020, mayo 8). *Métricas de Clasificación*. The Machine Learners.

<<https://www.themachinelearners.com/metricas-de-clasificacion/>>

[19] Parra, F. (2022, Septiembre 1). *6 Métodos de clasificación*. Bookdown.org.,

<<https://bookdown.org/content/2274/metodos-de-clasificacion.html>>

[20] Rodrigo, J. A. (s/f). *Regresión logística simple y múltiple*. Cienciadedatos.net. (Septiembre, 2022)

<https://www.cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple>

[21] Wikipedia contributors. (Setiembre, 2022). *Random forest*. Wikipedia, The Free Encyclopedia.

<https://es.wikipedia.org/w/index.php?title=Random_forest&oldid=140501165>

[22] Pang-Ning, T., Steinbach, M., 2013. *Introduction to Data Mining*. Pearson New International Edition.

[23] Hsu, C.W., Lin, C.J., 2002. A comparison of methods for multiclass support vector machines. *IEEE Trans. on Neural Networks* 13 (2), 415–425.

Códigos

[24] Arriaga, Luis, Julio 2020. Código de TFG. Diseño e implementación de un sistema autónomo para la simulación de fugas en depósitos. Github.

<<https://github.com/LuisArriaga13/TFG-Luis-Arriaga-Campos>>

[25] Yanes, Nicolás, Septiembre 2022. Código de TFM. Diseño e implementación de un sistema autónomo para la detección de fugas en depósitos. Github.

<https://github.com/Nikoru94/TFM_Fugas_ML>