



**Universidad
Politécnica
de Cartagena**

Entregable 2

Sistema de Monitorización Logística

Álvaro Serrano Delmás

10/06/2025



ÍNDICE

1. Justificación de los patrones de diseño.....	2
2. Control de versiones.....	3
3. Repositorio online.....	6

1. Justificación de los patrones de diseño

La práctica pide que cada una de los siguientes requisitos se cumplan utilizando un patrón de diseño diferente. Los requisitos son los siguientes:

- R1: En primer lugar, debe de existir una única instancia del sistema que gestione todos los componentes y recursos del sistema logístico.
- R2: Cada camión notifica al sistema con un nuevo valor de localización junto a valores de humedad y temperatura, cada 5 segundos de forma que el sistema pueda recibir actualizaciones de datos en tiempo real y procesarlas adecuadamente.
- R3: Cada nuevo valor de localización recibido debe de implicar la realización de una serie de pasos encadenados. Dichos pasos son los siguientes:
 1. Calcular diferentes estadísticos de la temperatura y humedad (ej: media, desviación típica) durante los últimos 60 segundos.
 2. Luego, comprobar si la temperatura actual del camión está por encima de un umbral (dicho umbral puede fijarse por el estudiante).
 3. Finalmente, comprobar si durante los últimos 30 segundos la variación de temperatura o humedad ha aumentado o disminuido más de 2 grados centígrados.
- R4: los cálculos de la variación de humedad y temperatura se debe lanzar concurrentemente con subrutinas (Asyncio de Python).
- R5: Antes de procesar los datos en el servidor, se deben adaptar los valores de localización en coordenadas GMS (o GD) que producen los camiones a códigos OCL, para almacenar la información de localización en este formato.

Teniendo en cuenta las restricciones anteriores, se han tomado las siguientes decisiones han tomados las siguientes decisiones:

- Para crear una única instancia optamos por un patrón Singleton en el que la sola instancia sea el sistema de gestión llamado Empresa.
- Para que los camiones vayan notificando los datos, el patrón que nos va a servir va a ser el de Observer. Así los propios camiones pueden notificar a sus suscriptores (la propia empresa) sus actualizaciones en los datos.
- Para que cada gestión se realice de manera ordenada y encadenada optamos por chain of responsibility. Así nos permitirá hacer una de las funciones y después otra en el orden que nosotros establezcamos. Cada una de las funciones a realizar se desarrollará en una clase que heredará de un Manejador genérico.

- La forma de implementar las llamadas concurrentes es involucrar Asyncio en el patrón Observer. De esta manera este patrón se ejecuta cada 5 segundos para que se simule la actualización en tiempo real de los camiones.
- Para adaptar los valores de un tipo de coordenadas a otras usamos un patrón Adapter y así guardar los datos en el formato deseado.

Más cosas a remarcar de la implementación es la creación de las clases Camion o Coordenadas que se abstraen para facilitar el funcionamiento del programa.

Por otra parte, las funciones de segundo orden las implementamos en los manejadores y el adaptador que son realmente los momentos en los que se manipulan realmente los datos.

Para los test hemos decidido implementar un test que compruebe cada patrón de diseño y además un test que analice cada excepción añadida en la implementación. Esto lo que da como resultado 6 tests que comprobarán la eficacia y validez de nuestro código.

2. Control de versiones

Primero empezamos, inicializando nuestro repositorio a nivel local.

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git init
Initialized empty Git repository in C:/Users/Álvaro/OneDrive - Universidad Politécnica de Cartagena/Escritorio/pcd/entregable2/.git/
```

Después de configurarlo para asociarlo al repositorio online, añadimos nuestros archivos del entregable.

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git add implementacion_transp
orte_mercancias_alvaro_serrano.py

C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git commit -m "first commit"
[master (root-commit) e1b0ad4] first commit
1 file changed, 32 insertions(+)
create mode 100644 implementacion_transporte_mercancias_alvaro_serrano.py

C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git branch -M main

C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git remote add origin https://
github.com/alvaro-serrano-upct-es/pcd_entregable2_AlvaroSerrano.git
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git push -u origin main
fatal: unable to access 'https://github.com/alvaro-serrano-upct-es/pcd_entregable2_AlvaroSerrano.git/': Could not resolve host: github.com

C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 609 bytes | 55.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/alvaro-serrano-upct-es/pcd_entregable2_AlvaroSerrano.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Teniendo los archivos subidos, los cambios que se realicen en estos durante el desarrollo, se llevarán a cabo en una rama paralela la cual creamos con el nombre de development.

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git checkout -b development
Switched to a new branch 'development'

C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git status
On branch development
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    "dise\303\261o_uml.uxf"
    entregable-2-PCD_2425_upct.pdf
    gestion de versiones/
    gms-to-olc.py

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git branch
* development
  main
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git push origin development
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 2.93 KiB | 1.46 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/alvaro-serrano-upct-es/pcd_entregable2_AlvaroSerrano.git
 bbbca18..ef2e574  development -> development
```

Cuando lleguemos a un punto en el cual queremos fusionar las ramas usamos merge.

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git checkout main
M   tests_AlvaroSerrano.py
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git branch
* main
  development

C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git merge development
Updating bbbca18..ef2e574
Fast-forward
 ...ntacion transporte mercancías alvaro serrano.py | 226 ++++++-----
 1 file changed, 181 insertions(+), 45 deletions(-)
```

Y además en los puntos en los que nos interesa guardar la versión con una etiqueta específica lo hacemos con un tag.

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git tag implementacion_inicial
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git tag
implementacion_inicial
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git push origin implementacion_inicial
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/alvaro-serrano-upct-es/pcd_entregable2_AlvaroSerrano.git
 * [new tag]          implementacion_inicial -> implementacion_inicial
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git tag implementacion_final
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git push origin implementacion_final
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/alvaro-serrano-upct-es/pcd_entregable2_AlvaroSerrano.git
 * [new tag]          implementacion_final -> implementacion_final
```

Cuando vayamos haciendo cambios pequeños los vamos actualizando de la misma manera con los comandos add, commit y push.

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git add implementacion_transporte_mercancias_alvaro_serrano.py
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git commit -m "Primera versión del código final"
[development ef2e574] Primera versión del código final
1 file changed, 181 insertions(+), 45 deletions(-)
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git add implementacion_transporte_mercancias_alvaro_serrano.py
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git commit -m "Actualizaciones y correcciones"
[main 554cb6e] Actualizaciones y correcciones
1 file changed, 3 insertions(+), 3 deletions(-)
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git push origin main
fatal: unable to access 'https://github.com/alvaro-serrano-upct-es/pcd_entregable2_AlvaroSerrano.git/': Could not resolve host: github.com
```

```
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd\entregable2>git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 403 bytes | 403.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/alvaro-serrano-upct-es/pcd_entregable2_AlvaroSerrano.git
ef2e574..554cb6e main -> main
```

Por último, también usamos la terminal para ejecutar nuestro pytest.

```

C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd>python -m pytest entregable2/tests_AlvaroSerrano.py
===== test session starts =====
platform win32 -- Python 3.13.1, pytest-8.4.0, pluggy-1.6.0
rootdir: C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd
collected 6 items

entregable2\tests_AlvaroSerrano.py FF.... [100%]

===== FAILURES =====
_____ test_raises_exception_on_Empresa _____

    def test_raises_exception_on_Empresa():
        s1 = Empresa.obtener_empresa()
        with pytest.raises(TypeError):
            s2 = Empresa()
            ^^^^^^^^^

entregable2\tests_AlvaroSerrano.py:12:
-----
self = <implementacion_transporte_mercancias_alvaro_serrano.Empresa object at 0x000001D4790EC2D0>

    def __init__(self):
        if Empresa._unicaEmpresa is not None:
            raise Exception("¡Esta clase es un Singleton! Utilice 'obtener_empresa()' para obtener la instancia.")
E           Exception: ¡Esta clase es un Singleton! Utilice 'obtener_empresa()' para obtener la instancia.

entregable2\implementacion_transporte_mercancias_alvaro_serrano.py:154: Exception
_____ test_raises_exception_on_matricula _____

    def test_raises_exception_on_matricula():
        with pytest.raises(TypeError):
            Camion("C1")

entregable2\tests_AlvaroSerrano.py:17:
-----
self = <implementacion_transporte_mercancias_alvaro_serrano.Camion object at 0x000001D47EC72900>, matricula = 'C1'

    def __init__(self, matricula: str):
        if len(matricula) != 7:
            raise Exception("La matrícula debe tener 7 caracteres.")
E           Exception: La matrícula debe tener 7 caracteres.

entregable2\implementacion_transporte_mercancias_alvaro_serrano.py:128: Exception
===== short test summary info =====
FAILED entregable2/tests_AlvaroSerrano.py::test_raises_exception_on_Empresa - Exception: ¡Esta clase es un Singleton! Utilice 'obtener_empresa()' para obtener la instancia.
FAILED entregable2/tests_AlvaroSerrano.py::test_raises_exception_on_matricula - Exception: La matrícula debe tener 7 caracteres.
===== 2 failed, 4 passed in 1.69s =====
C:\Users\Álvaro\OneDrive - Universidad Politécnica de Cartagena\Escritorio\pcd>

```

Podemos observar que los test que comprueban los patrones de diseño son superados con éxito y que los fallos obtenidos son de los test encargados de hacer saltar las excepciones del programa.

3. Repositorio online

Adjunto el enlace al repositorio online donde están guardados los archivos del entregable:

https://github.com/alvaro-serrano-upct-es/pcd_entregable2_AlvaroSerrano.git