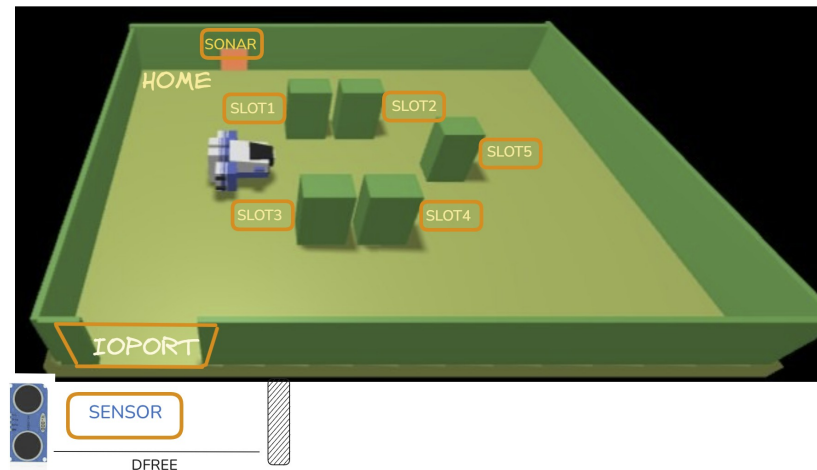


## TemaFinale25

A *Maritime Cargo shipping company* (from now on, simply *company*) intends to automate the operations of load of freight in the ship's cargo hold (or simply *hold*). To this end, the company plans to employ a *Differential Drive Robot* (from now, called *cargorobot*) for the loading of goods (named *products*) in the ship's hold.

The products to be loaded must be placed in a container of predefined dimensions and registered, by specifying its *weight*, within a database, by using a proper service (*productservice*). After the registration, the *productservice* returns a unique product identifier as a natural number *PID*,  $PID > 0$ .

The hold is a rectangular, flat area with an Input/Output port (*IOPort*). The area provides **4 slots** for the product containers.



In the picture above:

- The **slots** depict the *hold storage areas*, when they are occupied by *product containeres*
- The **slots5** area is **permanently occupied**, while the other slots are initially empty
- The **sensor** put in front of the *IOPort* is a sonar used to detect the presence of a product container, when it measures a distance *D*, such that  $D < DFREE/2$ , during a reasonable time (e.g. **3** secs).

### TF2025 Requirements

The company asks us to build a software systems (named *cargoservice*) that:

1. is able to receive the request to load on the cargo a product container already registered in the *productservice*.

The request is rejected when:

- the product-weight is evaluated too high, since the ship can carry a maximum load of **MaxLoad > 0 kg**.
- the hold is already full, i.e. the **4 slots** are already occupied.

If the request is accepted, the *cargoservice* associates a slot to the product *PID* and returns the name of the reserved slot. Afttwerds, it waits that the product container is delivered to the *ioport*. In the meantime, other requests are not elaborated.

2. is able to detect (by means of the *sonar* **sensor**) the presence of the product container at the *io-*

*port*

3. is able to ensure that the product container is placed by the *cargorobot* within its reserved slot. At the end of the work:
  - the *cargorobot* should returns to its **HOME** location.
  - the *cargoservice* can process another *load-request*
4. is able to show the current state of the *hold*, by mesans of a dynamically updated *web-gui*.
5. **interrupts** any activity and turns on a led if the *sonar sensor* measures a distance **D > DFREE** for at least **3** secs (perhaps a sonar failure). The service continues its activities as soon as the sonar measures a distance **D <= DFREE**.

### Before starting

- Un articolo del 2011 all'alba dei microservizi: [\*msolIEEE.pdf\*](#)
- Come si costruiscono oggi le applicazioni sulla base di microservizi: [\*Chiesto a Gemini \(28/05/2025\)\*](#)

### Il processo di pruduzione

#### *Processi agili e incrementali*

Introduzione a SCRUM e al concetto di SPRINT legato alla costruzione di prototipi estendibili

#### *DomainDrivenDesign*

Introduzione a un approccio alla progettazione e costruzione di software che pone l'enfasi sulla comprensione profonda del dominio di business

#### *TemaFinaleNote*

Indicazioni su come organizzare il lavoro finale