

# Task 2

## Report

Liubov Mikhailovna 60204 and Jorge Fresco 60209

### Concurrency

- **Concurrency is ensured by modeling each conveyor belt and truck as independent Erlang processes** using spawn, with all processes operating concurrently. The package generator (*generator\_loop*), conveyor belts (*conveyor\_loop*), and trucks (*truck\_loop*) work in parallel without interference.
- Each conveyor belt has an independent queue of packages received from the generator, communicating with trucks via message passing, while trucks maintain their own state (remaining space), accepting packages until full or out of space for the next package and replacing themselves instantly by resetting their remaining space.

### Deadlock-Free Operation

- The system avoids deadlocks by ensuring that all processes are non-blocking, with asynchronous message handling and making sure no packages are lost.
- **Conveyor belts** never wait indefinitely; they either periodically process their queue by sending a package to its truck to be loaded or buffer new received packages for future processing in their queue. If a package doesn't fit into the truck, it is retrieved and put back into the queue again to be processed later after the truck is replaced.
- **Trucks** are always ready to receive packages and automatically replace themselves when full or when the received package doesn't fit into the truck because its size is greater than remaining space. When it doesn't fit, the truck sends the package back to the conveyor belt to be put back in its queue. This ensures that all packages are eventually loaded into a truck and none are lost or destroyed.

## Progress Guarantee

- **Package generator** continuously creates packages, with a random size between 1 and 3, at a regular interval and sending them to a randomly chosen conveyor belt.
- **Conveyor belts** are always working either periodically processing their queue, sending packages to their assigned truck or receiving rejected packages from the truck.
- **Trucks** continuously handle incoming packages and replace themselves instantly when full or when out of capacity for the next package, in this case it sends the package back to the conveyor belt.
- This all ensures that all components are always active, and every package follows a logical flow of first being generated by the package generator, sent to a conveyor belt and finally loaded into a truck or if it doesn't fit into the truck, its put back in the conveyor belt to then be loaded into the next available truck.

## Message Passing

- All interactions between components or even to themselves occur via message passing:
  - **Generators → Conveyors:** {new\_package, Package}
  - **Conveyors → Trucks:** {load\_package, Package}
  - **Conveyors → Conveyors:** {process\_queue}
  - **Trucks → Conveyors:** {package\_rejected, Package}