# Task 1

## 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.
- Conveyor belts process their queues independently, communicating with trucks via message passing, while trucks maintain their own state, accepting packages until full and replacing themselves instantly. This guarantees efficient and independent operation of all components.

**Deadlock-Free Operation**

- The system avoids deadlocks by ensuring that all processes are non-blocking, with asynchronous message handling. Conveyor belts never wait indefinitely; they either process their queues ({process_queue}) or buffer packages for future processing.
- Trucks automatically replace themselves when full, ensuring conveyor belts are never left waiting for a truck to become available. Packages are either loaded immediately if space is available or retried by the conveyor if rejected.

**Progress Guarantee**

- The system guarantees progress by having the package generator (start_generator/2) continuously create packages at regular intervals and assign them to conveyor belts. Conveyor belts periodically process their queues and retry rejected packages until they are successfully loaded onto a truck.
- Trucks continuously handle incoming packages and replace themselves instantly when full, ensuring all components remain active and functional. Every package follows a logical flow:
    1. **Generated** by the package generator.
    2. **Buffered** on conveyor belts.
    3. **Loaded** onto trucks when space is available.
- This ensures that no package is left unprocessed or stranded indefinitely, maintaining continuous operation throughout the system.

**Message Passing**

- All interactions occur via message passing:
    - **Generators → Conveyors**: {new_package, Package}
    - **Conveyors → Trucks**: {load_package, Package}
    - **Trucks → Conveyors**: {package_rejected, Package}
- The message-passing mechanism ensures scalability, safety, and the absence of race conditions.