- The classes implemented are not meant to be subclassed. Therefore, they only consist of private and public methods and attributes.
- In order to maintain the original API of the Notification Channel, the new implementation was created in a way so that the logic of the program would remain the same, while incorporating the new underlying technology. To achieve this, new classes were added in order to compartmentalize the different tasks of the program.
    - To send messages through AMQP using the Qpid Proton library, there needs to be classes that subclass a proton::messaging_handler. For this purpose, two new classes were created: Sender and Receiver, for the Supplier and the Consumer, respectively. This classes serve with the sole objective of creating the channel and sending strings through them.
    - It was decided that to maintain a proper encapsulation and functionality of the program, Suppliers and Consumers will be the ones in charge to convert the structures to be sent into strings (or viceversa), while having a private instance of a Sender and a Receiver (respectively), which will serve only as a transport channel.
- Según la documentación de ActiveMQ:

AMQP and destinations

If an AMQP Link is dynamic then a temporary queue will be created and either the remote source or remote target address will be set to the name of the temporary queue. If the Link is not dynamic then the address of the remote target or source will be used for the queue. In case it does not exist, it will be auto-created if the settings allow.

AMQP and Multicast Addresses (Topics)

Although AMQP has no notion of "topics" it is still possible to treat AMQP consumers or receivers as subscriptions rather than just consumers on a queue. By default any receiving link that attaches to an address that has only `multicast` enabled will be treated as a subscription and a corresponding subscription queue will be created. If the Terminus Durability is either `UNSETTLED_STATE` or `CONFIGURATION` then the queue will be made durable (similar to a JMS durable subscription) and given a name made up from the container id and the link name, something like `my-container-id:my-link-name`. If the Terminus Durability is configured as `NONE` then a volatile `multicast` queue will be created.

Because of the above, it is required to have a created multicast address on the broker, where the dynamic queues can be created using AMQP links, via the proton library. These links (either sender or receiver) can be configured with a specific name using the sender or receiver options classes of Qpid Proton. They can be set to "dynamic" in the same way.