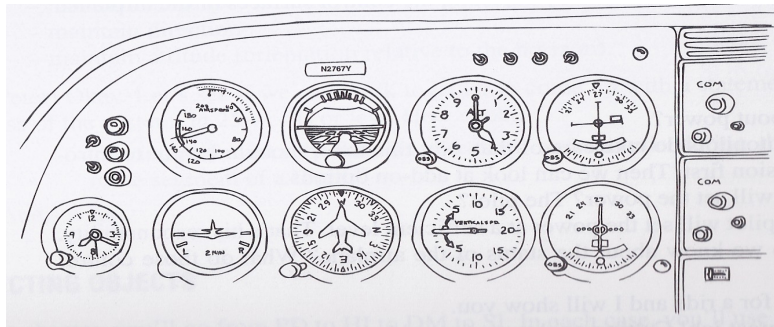
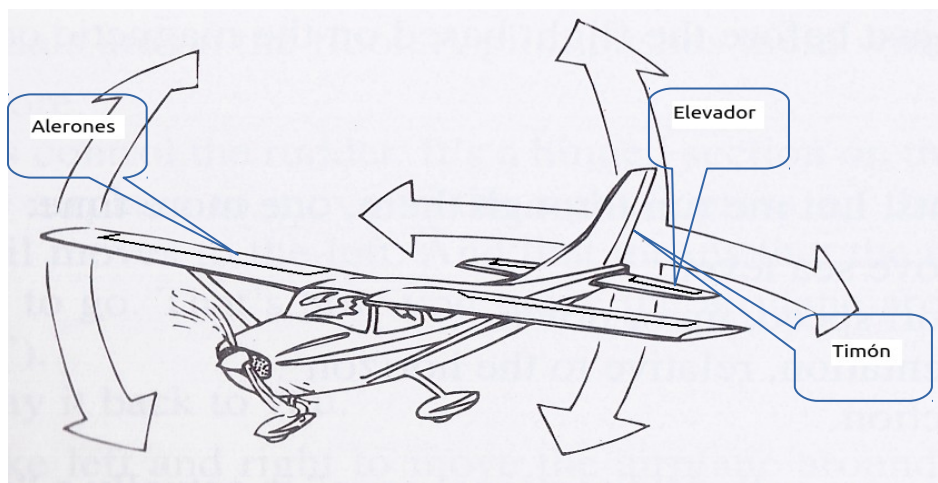


## Un problema de diseño de software

Mariano es piloto instructor de entrenamiento y nos ha consultado acerca de la posibilidad de dotar a su avioneta con un sistema de control automático de vuelo. Con el objetivo de conocer este negocio lo entrevistamos y nos dijo: estoy pensando en un sistema simple, que funcione en vuelo y no en despegue y aterrizaje. Nos explicó que una vez establecido el rumbo, la altura y la velocidad el sistema podría activarse y mantener las variables automáticamente. La nave tiene una pequeña computadora a la cual se le puede cargar software y ejecutarlo. Las variables a controlar serán medidas a través de los mismos sensores que utiliza el sistema original del avión para mostralas en los instrumentos de a bordo. Las acciones a efectos de modificar las variables serán ejecutadas por actuadores que ya tiene instalado el avión y a los cuales se puede acceder ya que presentan una interfaz con la computadora. Es decir que, simplificando el problema, para medir se debe leer una determinada posición de memoria y para actuar se debe escribir en otra posición de memoria.



Las variables que se miden y se muestran en los instrumentos son: el rumbo, en coordenadas norte/sur y este/oeste, la altura, la velocidad vertical la cual es medida por un giróscopo y presentada como tal en el indicador y la posición, la cual muestra el desvío del eje transversal que pasa por las alas o el eje longitudinal trompa/cola con referencia al horizonte. Las acciones para modificarlas se realizan así: el rumbo se modifica girando el volante de la nave para mover el timón a uno de sus lados; la altura se modifica tirando o empujando del volante de forma que los elevadores de la cola giren en torno a su eje o pisando uno de los dos patines (izquierdo o derecho) de manera que el avión se incline hacia el lado correspondiente y por último la posición, que se modifica al inclinarse el avión a uno de los lados o al elevarse o descender.



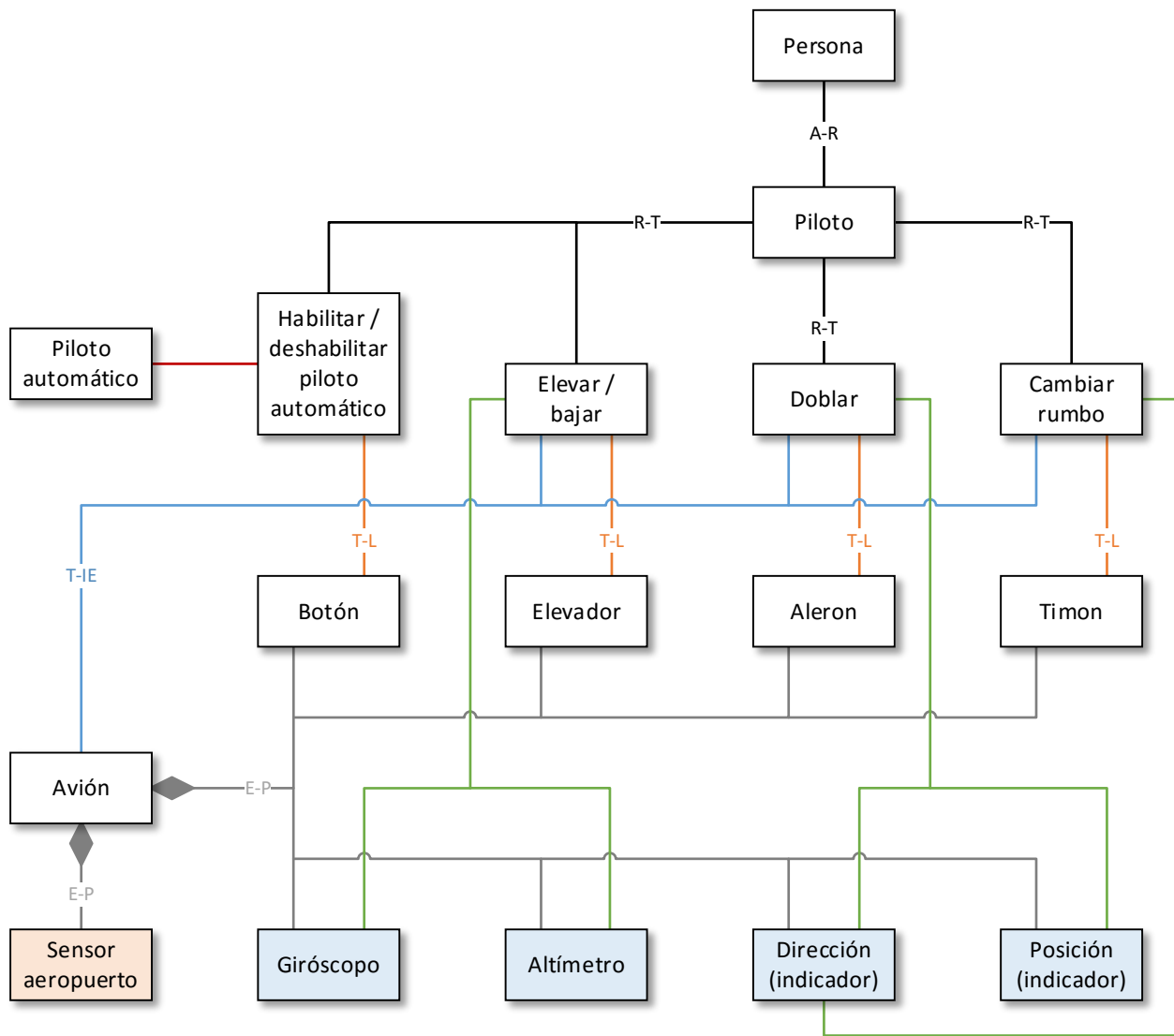
El sistema debe funcionar de manera que cuando se activa, tome de los sensores los valores de referencia de las variables y cuando se detecten cambios de las mismas se corrijan a partir de las acciones sobre los actuadores correspondientes. El canal de comunicación entre los sensores y la

computadora es único y a través del port correspondiente se reciben los datos provenientes de todos ellos. El stream de datos proveniente de estos sensores está formado por paquetes de datos cada uno de los cuales están precedido de un encabezamiento (código) que identifica al sensor.

En el tablero del avión hay botones libres, no asignados a ninguna funcionalidad, los cuales al ser presionados generan un evento que hace que el sistema operativo invoque a una función si se ha registrado alguna para tal fin. Otra fuente de información es un receptor automático que recibe la altura, velocidad y posición enviada desde un aeropuerto cercano solo cuando el avión se ha acercado lo suficiente como para iniciar maniobras de descenso.

Mientras relevábamos este problema se acercaron otros pilotos del aeroclub que mostraron interés en este sistema por lo cual se especula que se podrían vender más de uno, habrá que tener en cuenta esta cuestión ya que por lo que se pudo ver las avionetas tienen otras variables adicionales que se miden para componerlas en las variables mencionadas y diferentes marcas de sensores.

Se pide que usted desarrolle un modelo de dominio, defina una arquitectura preliminar y sugiera la utilización de criterios en el diseño del software basándose en la información de este relevamiento.



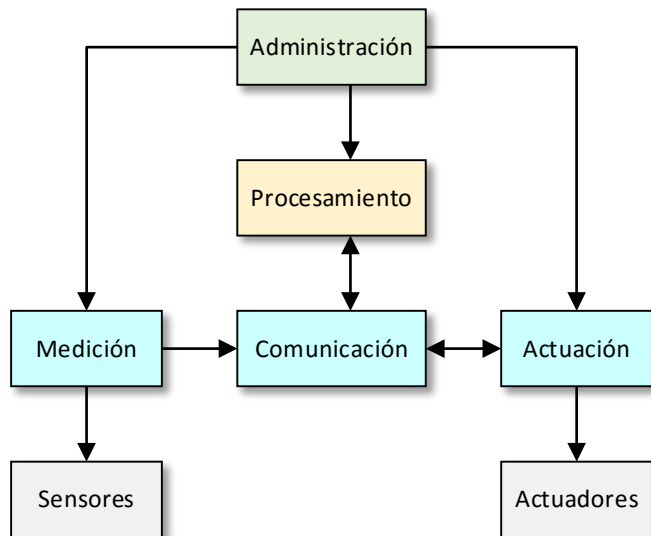
### Observaciones

- No todas las relaciones entre instancias son de algún patrón. No siempre se dan los patrones, pero debe estar igual la relación.
- Relación entre “Piloto automático” y “Habilitar / deshabilitar piloto automático” probablemente es T-R, y puede pasar que están faltando relaciones entre “Piloto automático” y sensores y actuadores del avión.
- “Sensor aeropuerto” funciona asincrónicamente, a diferencia de los otros 4.
- No hay interfaz del usuario: es el panel de avión.
- Pueden haber variantes de sensores o sensores nuevos: cambio de tecnología.

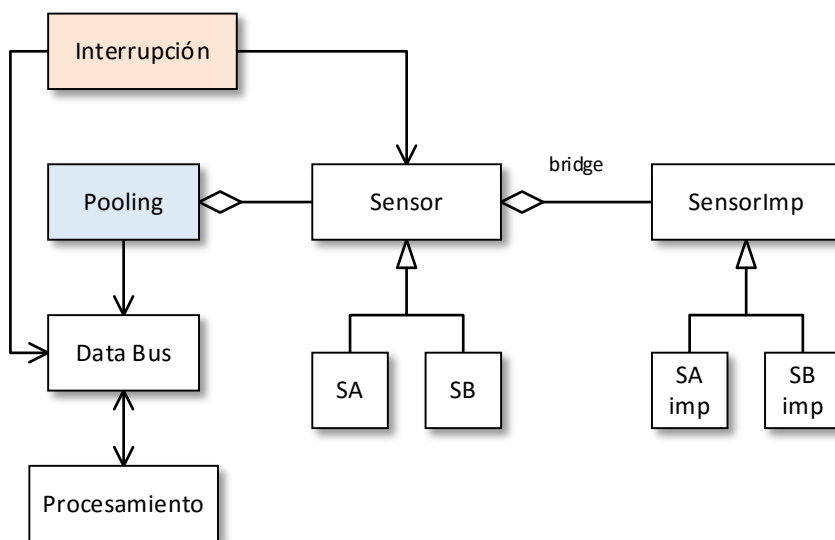
### Eligiendo arquitectura

- MVC no lo es ya que no hay vistas.
- Pipes and Filters no se usa en aplicaciones críticas, y acá es un avión en vuelo. Además las operaciones sobre el stream deben ser atómicas. Así que no va ese.
- Para aplicar MicroKernel deben haber ambas direcciones de cambio, pero acá solo cambia la tecnología (IS).
- El sistema no es distribuido: no sirve Broker.
- EA tampoco: hay un solo negocio, un solo usuario, no hace falta mapeo y administración de objetos y tablas, no se ve transaccionalidad, además es un sistema demasiado chico para estar implementado con EA.
- Layers es lo que queda...

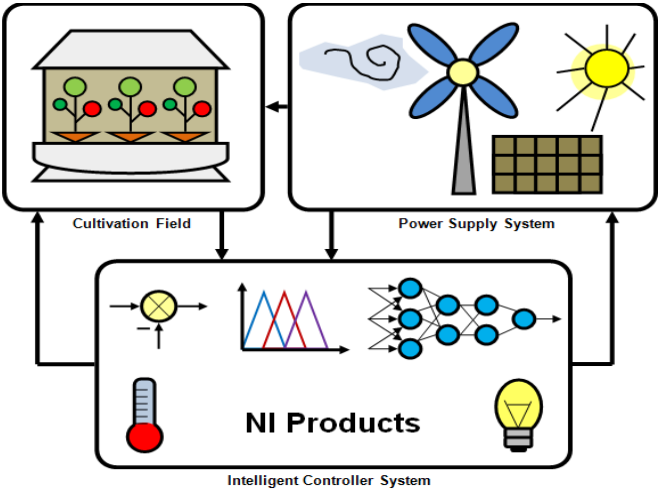
Cada color significa una capa, desde la superior (verde) de nivel muy alto hasta la inferior (gris) de nivel de hardware.



Hay que implementar sistema de muestreo sincrónico, por ejemplo Pooling (ya que dice que es una computadora chiquita, embebida, no tiene por qué soportar threads). Por otro lado para el tema de datos provistos por aeropuerto (lo cual solo es posible si está cerca) habría que utilizar un mecanismo de interrupción para toma de muestras asincrónica. Además hay distintos tipos de sensores y un montón de variantes de implementación de cada uno: conviene separar implementación de abstracción utilizando patrón de diseño Bridge.



Juana es la propietaria de un vivero cuya cantidad de clientes ha aumentado en los últimos tiempos. Esta razón le permite a ella interesarse ahora en instalar un sistema de software para el control de las variables fundamentales para que sus plantas mientras están en el depósito tengan las mejores condiciones de vida. Las variables a controlar son la humedad ambiente, la temperatura ambiente, la intensidad de la luz y el flujo de aire. Para esto Juana ha recibido información de parte de un proveedor de sensores para estas magnitudes que nos ha entregado. También le han provisto información de válvulas de humidificación por microgotas controladas eléctricamente así como equipos de aire acondicionado, ventiladores de potencia, luces y cortinas que se enrollan y desenrollan accionadas por motores eléctricos paso a paso.



Si bien los sensores aportan información de una única magnitud, las variables a controlar son una composición de las magnitudes medidas. Es decir que las acciones a tomar dependen de la información aportada por más de un sensor. La lógica de generación de acciones debe estar basada en las dependencias de las variables y las acciones sobre ellas como se muestra en la tabla

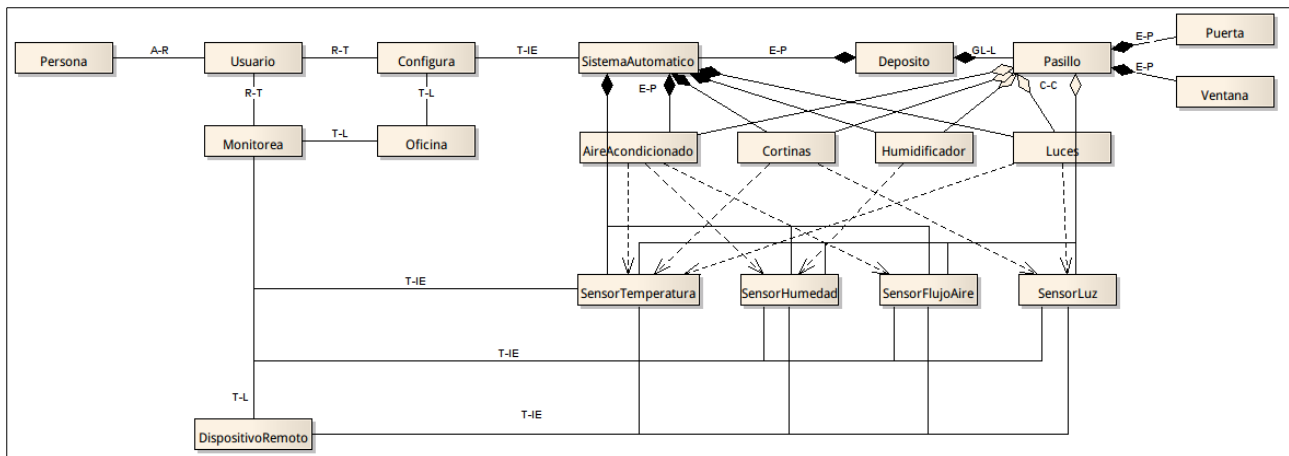
| Magnitudes medidas | Actuadores         |          |               |       |
|--------------------|--------------------|----------|---------------|-------|
|                    | Aire acondicionado | Cortinas | Humidificador | Luces |
| Temperatura        | X                  | X        |               | X     |
| Humedad            | X                  |          | X             |       |
| Flujo Aire         | X                  |          |               |       |
| Intensidad Luz     |                    | X        |               | X     |

Los sensores y actuadores estarán ubicados en los pasillos del depósito, los cuales podrán contar con uno o más de cada uno de estos componentes de acuerdo al tipo de plantas para el cual fue diseñado el pasillo. La parametrización de los componentes debe realizarse sobre cada uno de los sensores y actuadores. Hoy día se piensan instalar los actuadores mencionados, sin embargo es posible que en corto tiempo se instalen nuevos por lo que es importante que el diseño resuelva la comunicación de las acciones a los actuadores, una vez desarrolladas las clases que las modelan, permitiendo incorporar los nuevos con el menor esfuerzo de desarrollo (sin modificar el mecanismo de transmisión de acciones). Debe resolverse el canal de comunicación de forma de agregar nuevos componentes de hardware y el software que los modela sin generar cambios importantes en el diseño.

Se piensa instalar este software en una PC insdustrial con un sistema operativo eficiente y simple por lo cual las interfaces de usuario serán simples y pocas, debiendo permitir la configuración de los parámetros del sistema, su activación y desactivación y el monitoreo de las variables a modo de tablero de control desde la oficina contigua al depósito donde esté alojada la PC. Por lo que se pudo relevar hasta el momento se sabe que hay dos tipos de sensores, unos pasivos y otros activos por lo cual el diseño del software deberá tener en cuenta a ambos tipos. Necesitamos construir un modelo de dominio, bosquejar la arquitectura preliminar e indicar la posibilidad de utilizar patrones en el diseño.

## Solución

### Modelo de dominio



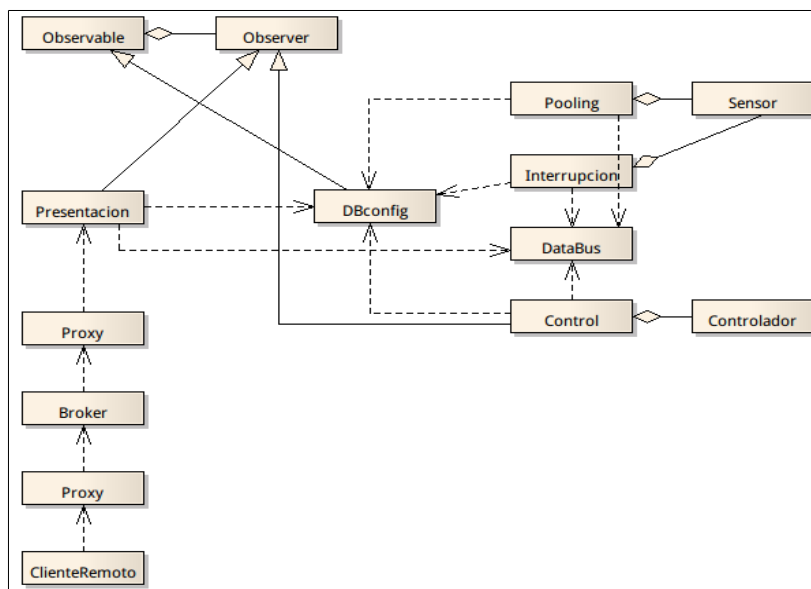
En la figura se indican los patrones de colaboración utilizados. La relación entre las magnitudes a medir y los actuadores se indica con flechas de dependencia, esto deberá ser tenido en cuenta al momento de implementar el algoritmo de control, tema fuera de consideración de este ejercicio.

El sistema automático en desarrollo está compuesto de una cantidad determinada de sensores y actuadores que pueden ser ubicados y reubicados en diferentes lugares de los pasillos.

El monitoreo puede hacerse en forma local (en la oficina) o remota; la configuración solo en forma local.

### Arquitectura propuesta

La arquitectura de este sistema tiene un componente de gran incidencia que es el aspecto de tiempo real. Las magnitudes a medir poseen una frecuencia de cambio baja por lo cual se decidió utilizar Pooling para los sensores de (temperatura, humedad, flujo de aire e intensidad de luz); sin embargo la apertura de aberturas obliga a utilizar sensores de luz y flujo de aire en modo Interrupción ya que estas acciones generan cambios bruscos en estas magnitudes, no así en la humedad ambiente y temperatura. Los sensores al igual que los actuadores intercambian información a través de un Databus.



Para satisfacer el requerimiento de monitoreo remoto se decidió utilizar un Broker que en la figura se indica con sus proxys correspondientes. Para que los cambios del modelo almacenados en una base de datos y en el Databus se reflejen en las vistas del monitoreo se incorporó un MVC con su mecanismo de propagación de cambios (Observable, Observer) como se muestra en la figura. La flexibilidad del sistema en lo referente a cambios en la incorporación de otro tipo de sensores se facilitará en el diseño del mismo utilizando patrones que permitan desacoplar las implementaciones como Bridge. La afectación de la arquitectura con Microkernel es un esfuerzo no justificado y que no aplica ya que hay una sola dirección de cambio.

## Ejercitación: trabajo con requerimientos



Se desea desarrollar una aplicación que corra en un teléfono celular marca BB para la carga de datos de inspecciones de automotores para una empresa de seguros. La empresa posee una dotación de 100 agentes inspectores. Cada agente de seguros, portador del teléfono que la empresa le entregó recibirá a través de la aplicación la agenda del día con los datos de los automotores a inspeccionar. Cada agenda está identificada con el número de empleado del agente el cual consta de veintiuno caracteres alfanuméricos, este será verificado al momento de ser recibida. La agenda posee los datos de cada automotor a inspeccionar y los de su propietario. Las agendas son diarias y cada una no podrá incluir más de 6 inspecciones, además la aplicación no podrá recibir una agenda si previamente no envió a la empresa todas las inspecciones de la agenda del día anterior.

El agente se traslada al domicilio del propietario y realiza la inspección del automotor agendado para ser asegurado.

Al realizar cada inspección el agente deberá registrar la marca, el modelo y los números de serie de cada componente (chasis, ruedas, motor, carrocería) y verificar y registrar su estado de conservación.

Cada componente podrá estar en buen, regular o mal estado. En el caso de no poder ser identificado deberá marcarse como no legible.

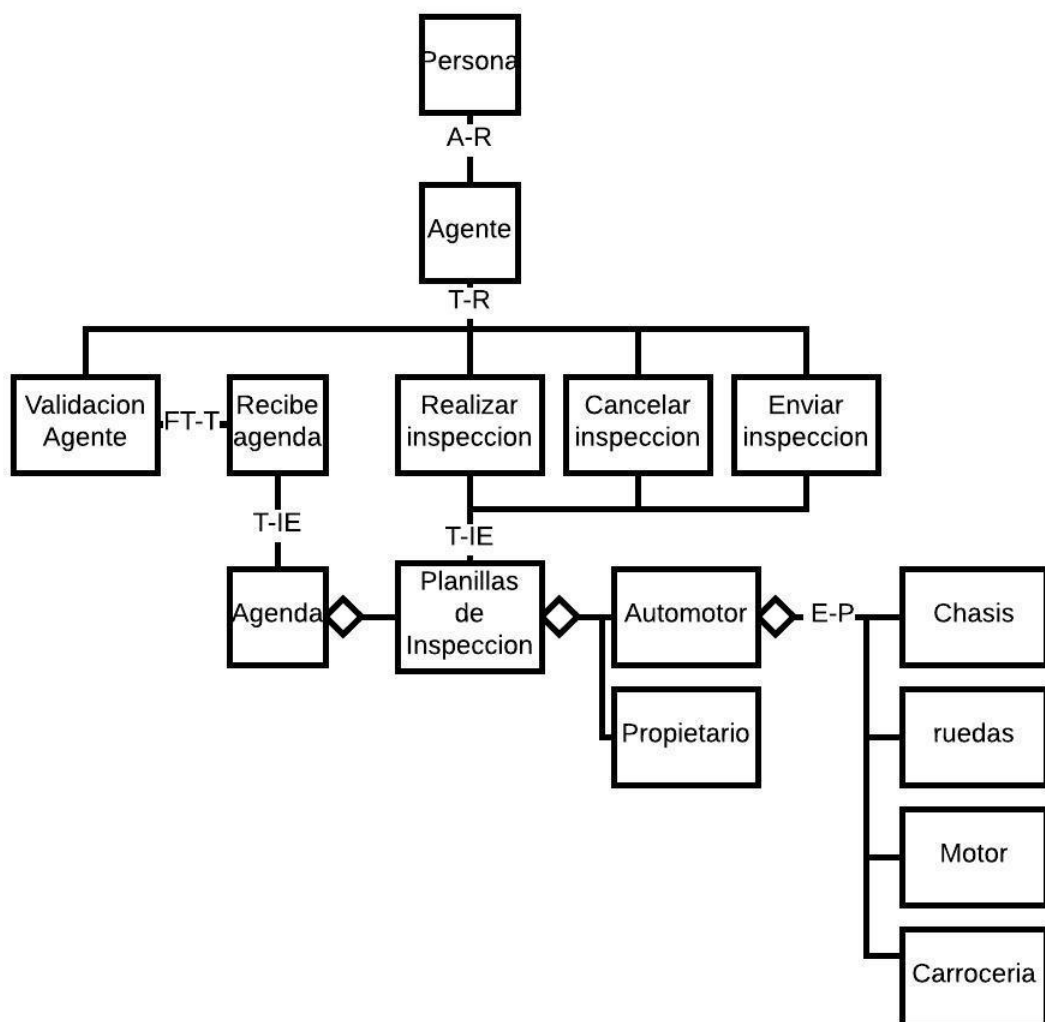
Las inspecciones podrán realizarse de una vez o podrán ser guardadas para ser terminadas en otro momento. Las inspecciones podrán ser canceladas y también marcadas como de imposible realización registrando la razón. Una vez completada una inspección, esta podrá ser enviada a la empresa o almacenada para ser enviada a posteriori. Las inspecciones no realizadas o canceladas no podrán ser enviadas a la empresa.

Nos encargaron hacer un modelo de dominio a efectos de validar las reglas de negocio para ser utilizado en el posterior diseño.

Trabajo a realizar: relevar requerimientos, bosquejar casos de uso y construir un modelo de dominio.

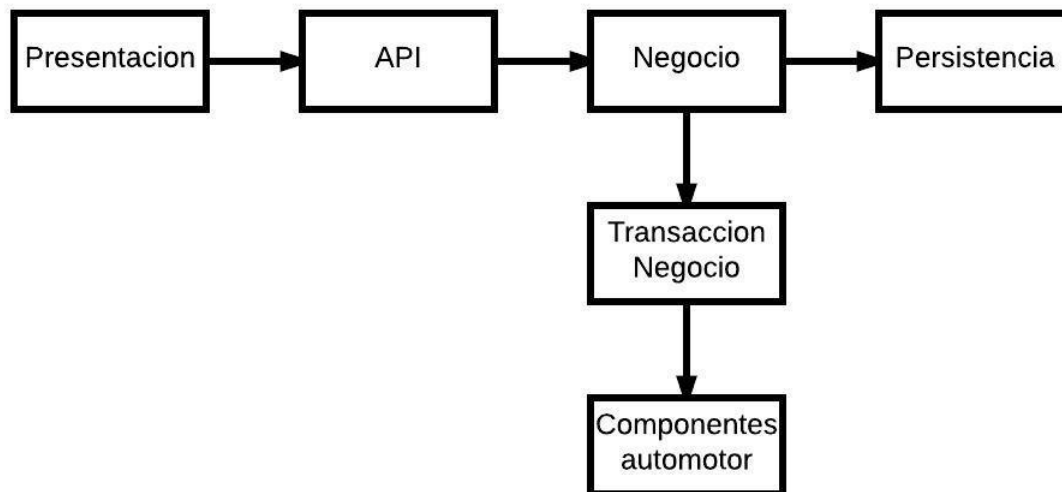


## Modelo de Dominio



## Arquitectura

Patrones: Cliente servidor, MVC para  
cambio de interfaces y en el negocio



## Ejercitacion

Una empresa de desarrollos electrónicos planea lanzar al mercado el instrumento universal de diagnóstico que ha bautizado con el nombre de “RobotDoc”. Para este proyecto nos ha contratado el desarrollo del software y nos ha pedido una primera impresión.

El aparato será una especie de robot ambulante y tendrá como parte de su arquitectura una computadora donde se ejecutará el software, una carcasa que le dará usabilidad desde el punto de vista ergonómico y un componente electrónico que permite conectarse con otros instrumentos que no poseen medios de comunicación estándar como las computadoras (RJ45, USB, etc.). Está orientado a centros de salud que no poseen un sistema integrado de información.

Se pensó a “RobotDoc” como un recolector de información de los diferentes consultorios/laboratorios donde se realizan los siguientes estudios: electroencefalograma, electrocardiograma, resonancia magnética, rayos X y electromiograma. En estos casos el aparato se conecta con otros instrumentos y consume datos tales como señales muestreadas. En el caso de los laboratorios se conecta con las computadoras y consume datos de informes.

El operador se desplaza a cada lugar y al tomar los datos ingresa información del paciente de manera que al final del recorrido por diferentes áreas del hospital queda agrupada la información por paciente. Es decir que para cada paciente habrá un conjunto de estudios con los cuales los diferentes médicos que lo atiendan podrán elaborar un diagnóstico utilizándola.

“RobotDoc” es compartido por varios médicos, los cuales acceden a él y elaboran su diagnóstico con el enfoque de su especialidad. Para este fin “RobotDoc” debe presentar la funcionalidad de “diagnosticar” con enfoques preestablecidos para cada especialidad (gastroenterología, cardíaco, clínico, neurológico, etc.). Los diagnósticos no podrán ser realizados si no fue recolectada la información de todos los estudios solicitados. Aquellos estudios que daten de más de 15 días deberán repetirse, es decir que quedan invalidados para ser utilizados en la elaboración de un diagnóstico, solo serán usados como informativos. Cada paciente tendrá una historia clínica conformada por todos los estudios realizados y los diagnósticos realizados.

La información de los diferentes estudios deben poder relacionarse a partir de un conjunto determinado de algoritmos que aún no fueron entregados. Esta funcionalidad es una especie de asistencia al médico que permiten detectar diferentes fenómenos, medir distintas magnitudes, correlacionar sucesos de diferentes estudios, procesar señales e imágenes.

Los diagnósticos incluirán posibles causas, explicaciones, recomendaciones y prescripciones realizadas por el médico especialista en base a la información recolectada y procesada. Los profesionales para utilizarlos deben registrarse previamente y serán agrupados por el administrador según su especialidad. Las secretarías de las áreas del hospital podrán acceder a los estudios e imprimirlos para ser entregados a los pacientes. Todos los usuarios deben estar agrupados a efectos de poder acceder al uso.

Se cree que la experiencia compartida por profesionales de la salud con diferentes visiones y sugerencias respecto de las interfaces de usuario así como la aparición de nuevos instrumentos a conectar generará al año de uso la producción de un segundo release del software que nos encargan.

Toda la información de “RobotDoc” se almacena en un servidor del hospital, aunque este es el límite de nuestro contrato, es decir no incluye el desarrollo de ninguna parte de este servidor.

Hemos decidido elaborar, para presentar a la empresa, un Modelo de Dominio que muestre el funcionamiento de este negocio para validarlo y una definición preliminar de la Arquitectura donde se justifique cada una de las partes incluidas.

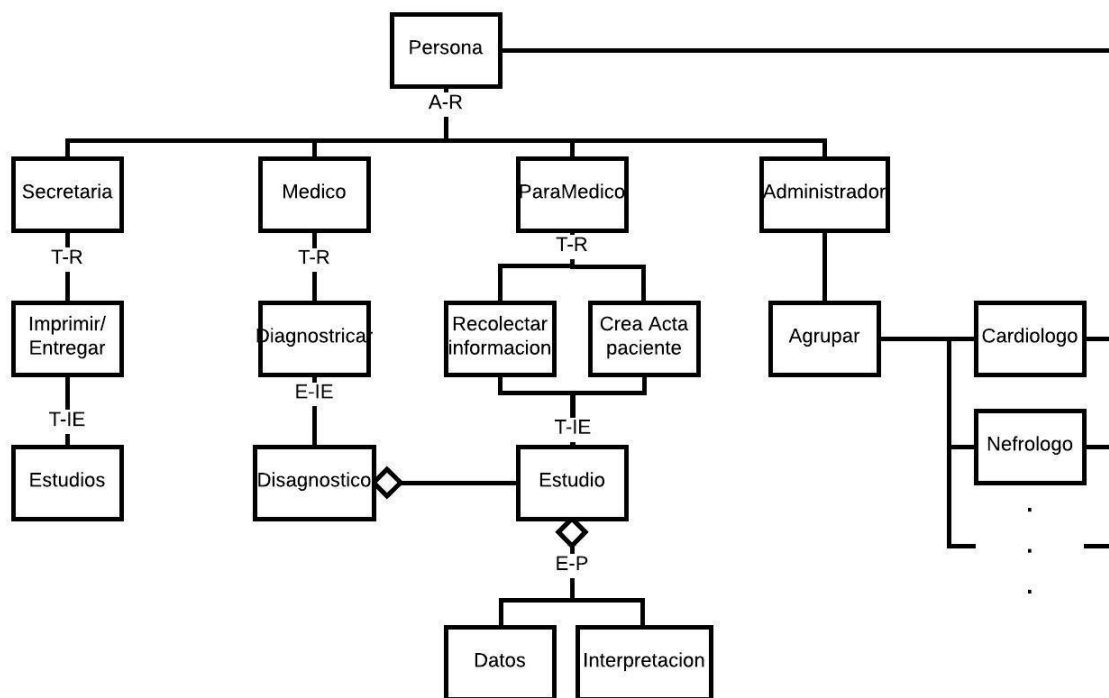
Un año después ... hay que agregar la funcionalidad de realizar diagnósticos con enfoque de medicina del deporte para lo cual es necesario conectarse con un aparato de electrocardiogramas de esfuerzo diferente a todos los manejados hasta ahora. También incluiremos los cambios sugeridos a algunas de las interfaces de los diagnósticos por imágenes. Por favor indique en que lugar de la arquitectura definida anteriormente impactarán estos tres cambios.

Pautas

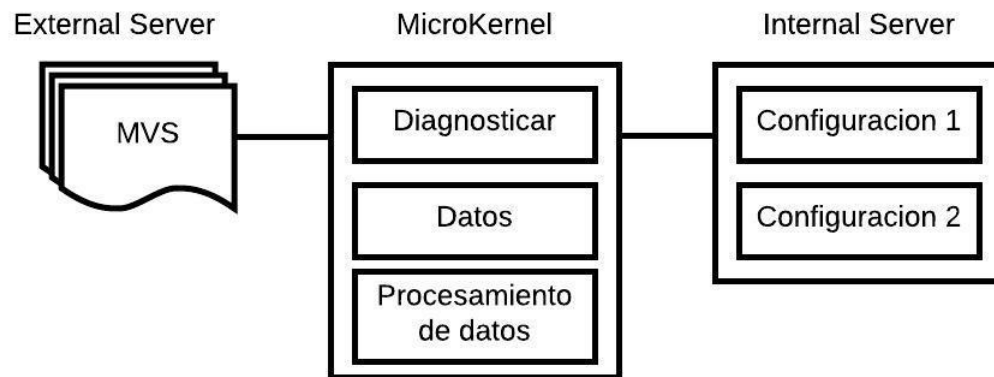
Sea breve y no escriba texto innecesario

Haga los diagramas que considere necesario

## Modelo de Dominio



## Arquitectura



Marta es la propietaria de un comercio de venta de kiosco y afines que ha crecido y ha sumado a su oferta bebidas, artículos de pastelería, revistas y diarios. Es un negocio de los llamados Polirubros. Este crecimiento ha llevado a Marta a interesarse en ordenar su negocio para incorporar tres cajeros en un local de mayores dimensiones. En este escenario Marta nos consulta acerca del desarrollo de un software que administre estos puntos de venta, para lo cual debemos construir un modelo de dominio y definir una arquitectura.

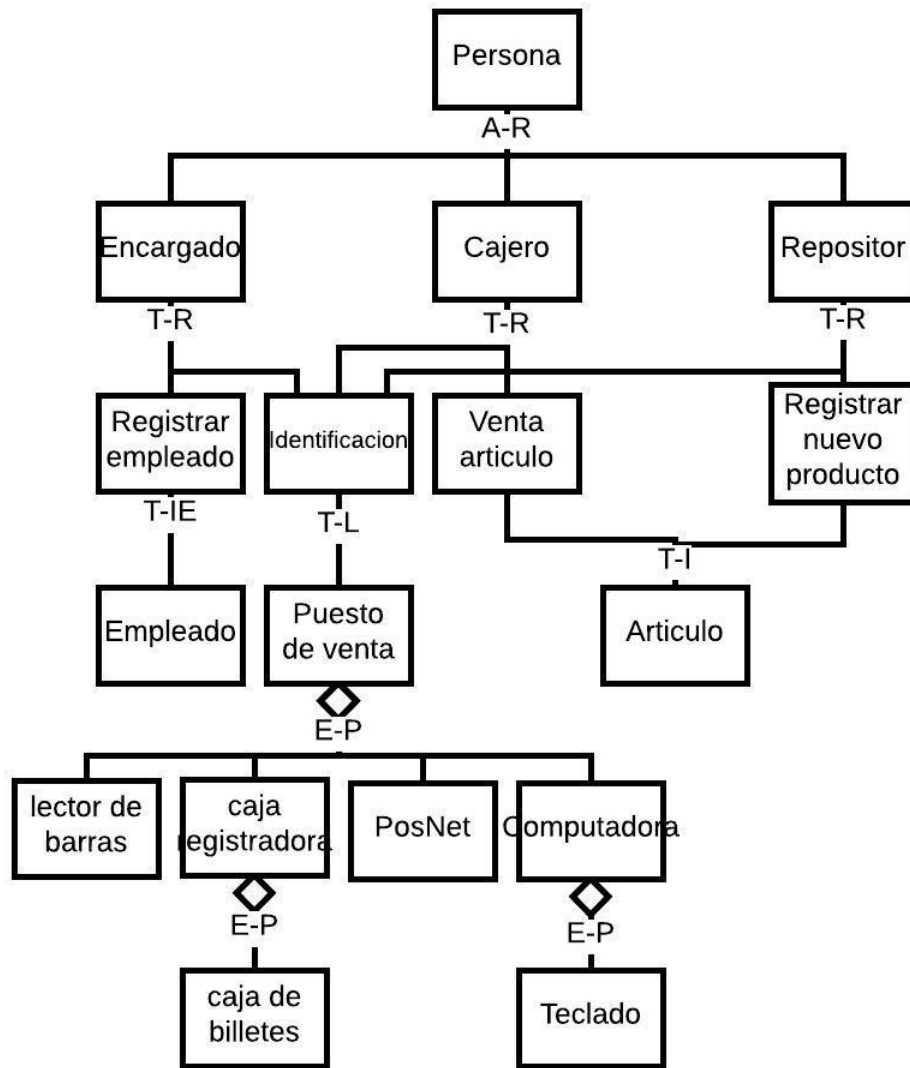


Cada puesto deberá ofrecer la posibilidad de que el empleado que lo ocupe se identifique para que el puesto de venta esté habilitado para su funcionamiento. Para esto Marta deberá previamente registrar a sus empleados. Cada puesto contará con una caja registradora con su caja porta billetes, su lector de barras y una conexión a la base de datos de productos. El software a desarrollar deberá administrar la dinámica de todos estos componentes además de registrar la venta de múltiples artículos y calcular los subtotales, totales e impuestos. Para el registro de los nuevos productos el sistema debe ofrecer a Marta la posibilidad de ingresarlos utilizando el lector de barras y la edición utilizando el teclado de la computadora de los puntos de venta. Marta piensa designar una persona con el rol de repositor para esta tarea. Por el momento a Marta no le interesa que el sistema realice el control de stock, solo sirva para hacer eficiente el cobro de los productos que sus clientes compren y se eviten las largas colas que desalientan a los clientes a ingresar al comercio.



Con esto en mente desea además que el sistema permita el pago de las ventas por medio de tarjeta de credito, con lo cual el sistema deberá incluir un dispositivo la posibilidad de validar dichas tarjetas. Este dispositivo formará parte de la caja registradora del punto de venta. El prestador del servicio de validación ya ha entregado la documentación de la forma de comunicación necesaria para el servicio, esta consiste en una comunicación asincrónica sobre un protocolo de alto nivel. Una cuestión adicional surgió en la última reunión relacionada a una conexión con el sistema contable que Marta ya posee, las ventas realizadas deberán impactarse en dicho sistema, el cual toma los registros de archivos con formatos especiales del sistema de archivos. Este tipo de desarrollo es nuevo para nosotros por lo cual deberíamos diseñar el software para que soporte cambios ante futuros pedidos o modificaciones.

## Modelo de Dominio



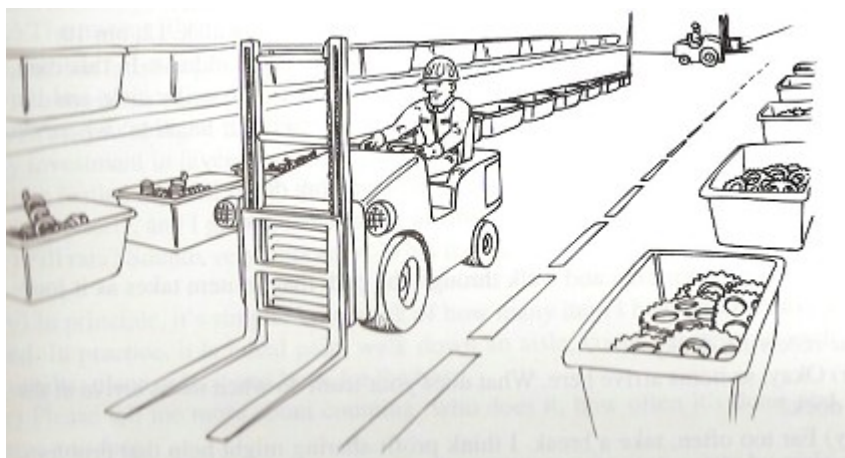


Figura 1 – Dársena de descarga

Actualmente el depósito de la empresa XXX recibe productos (ítems) de sus proveedores y los almacena hasta que reciben una orden de compra y deben armar el delivery (logística) de dichas órdenes. Hay una dársena de descarga (figura 1) donde arriban los pallets conteniendo los ítems que deben ser almacenados en tachos. Los operadores tratan de no mezclar distintos ítems en los tachos y además han ordenado los tachos en los pasillos del depósito de manera que les resulte cómodo buscar los ítems para las órdenes de compra que deben armar y despachar. Realizan todo el trabajo a mano, por ejemplo almacenan los pallets en tachos ubicados cerca de la salida según se reciben ya que es probable que estén asociados a un pedido para satisfacer una orden de compra que está por llegar. No poseen ningún sistema para llevar el control de existencia de los diferentes ítems, ni del almacenamiento en los tachos. Cada día un par de empleados realizan la cuenta de los ítems en stock.

Cuando necesitan armar un despacho para una orden de compra arribada, recorren los tachos buscando los ítems que necesitan, sin certeza de encontrarlos. Cada orden que no se despacha por falta de ítems es una pérdida. Tampoco desean acumular stock ya que esa forma de operar requiere de una inversión a veces parada e innecesaria.

El sistema que se desarrollará debe supervisar la recepción de productos facilitando, a los operadores, la descarga y depósito de los ítems recibidos en tachos con espacio libre a partir de la impresión de una guía de descarga con las coordenadas del lugar en donde realizar la descarga de acuerdo al control de cantidades en los tachos del depósito.

Por otro lado, para realizar la carga de los pallets a despachar a sus clientes, los ítems una vez empaquetados son acompañados del remito correspondiente generado en el depósito con los datos de los clientes que se han registrado en el sistema de ventas de la empresa.

Para esto se ha decidido incorporar a la planta una cinta transportadora que lleve los pedidos ya armados manualmente a las distintas bocas de expendio donde los



camiones distribuidores los cargarán para distribuirlos desde la dársena de carga (figura 2).

La idea es que cuando se reciben las órdenes y se vayan a imprimir como guía de armado de los pallets, se le asigne una boca y se le pegue un ticket autoadhesivo al pallet de forma que cuando el operador encargado de armarla apila todos los items en un pallet, lo deposita en la cinta y esta la transporta a la boca pre definida. Para esto al costado de la cinta hay un sensor detector que frena la cinta cuando arriba cada pallet para que un scanner lea el código asociado a la boca destino. Cada boca posee un sensor de posición para detectar el arribo de la carga y un mecanismo de empuje para posicionar el pallet frente a la boca de carga.

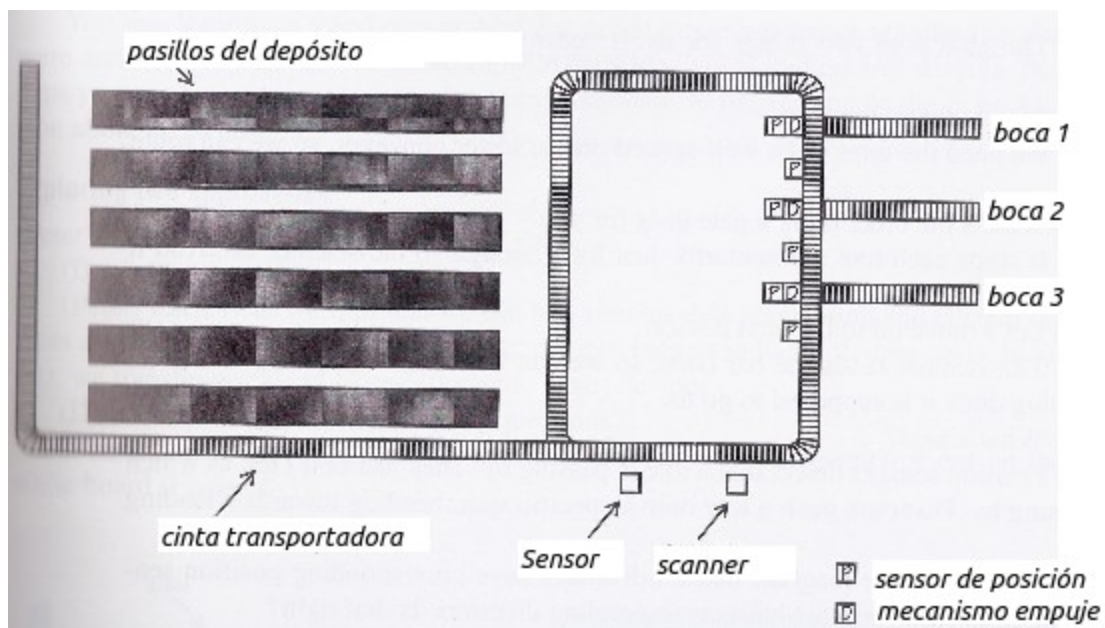
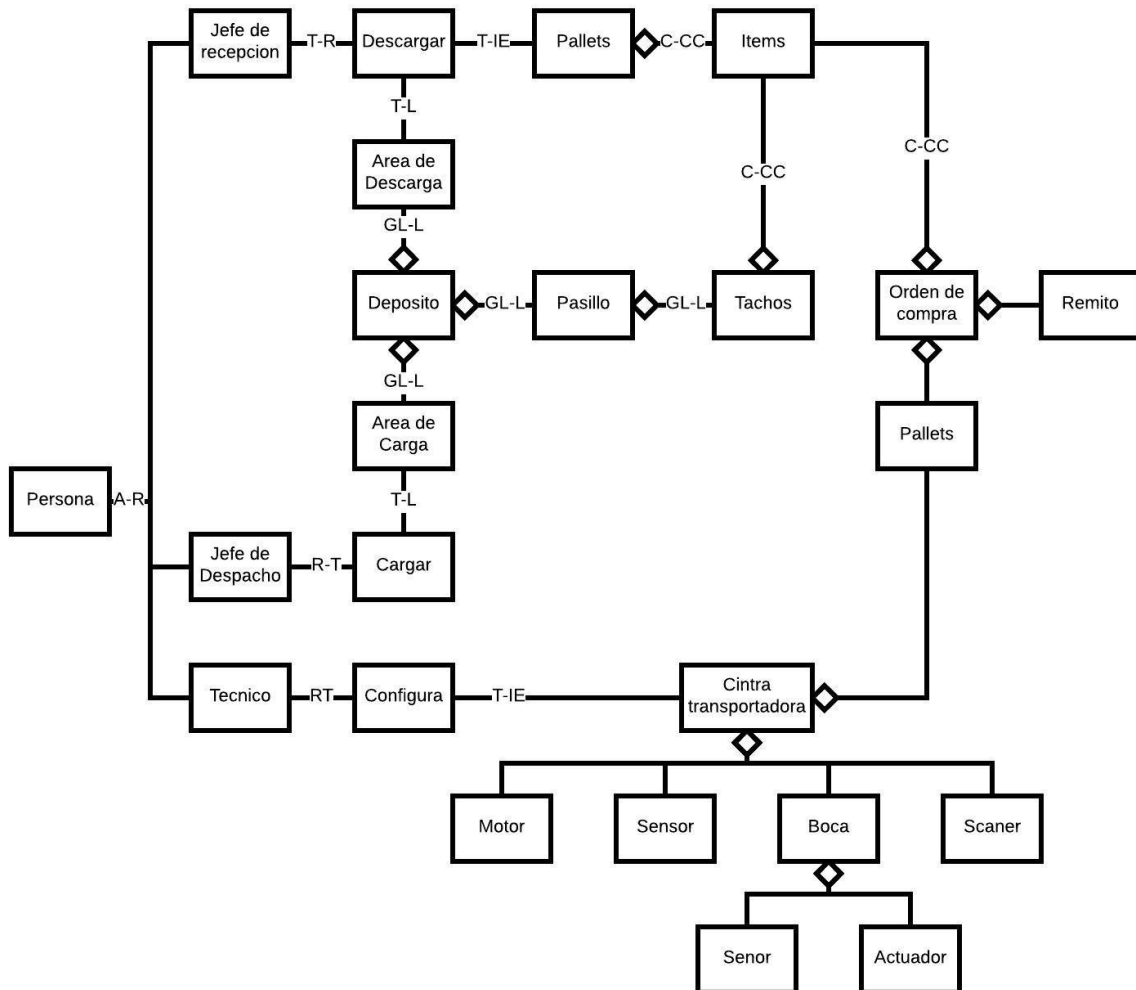


Figura 2 – Dársena de carga

Se está evaluando una alternativa a este sistema en el que cada mecanismo de empuje cuenta con su propio scanner, sin embargo dicho producto esta en la fase experimental por lo que por el momento se instalará el primero, no descartando reemplazarlo cuando se cuente con datos de buen funcionamiento de esta variante.

El sistema que se debe desarrollar debe contemplar la funcionalidad de la configuración de esta parte del sistema por parte del personal técnico; además debe incluir la funcionalidad de gestionar la descarga de los embarques recibidos por parte de un jefe de recepción y la funcionalidad de gestión de los embarques a despachar por parte de un jefe de despachos.

## Modelo de Dominio



## Arquitectura

