



UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

**SISTEMA DE INSPECCIÓN DE CARRILES: CONFIGURACIÓN Y
CÁLCULO DIMENSIONAL**



PEDRO MANSO BERNAL

JULIO 2014



UNIVERSIDAD DE OVIEDO

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN

MÁSTER EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE MÁSTER

**SISTEMA DE INSPECCIÓN DE CARRILES: CONFIGURACIÓN Y
CÁLCULO DIMENSIONAL**

DOCUMENTO II

DISEÑO GENERAL



PEDRO MANSO BERNAL

JULIO 2014

**ÁREA DE ARQUITECTURA Y
TECNOLOGÍA DE COMPUTADORES**

**TUTOR: DANIEL F. GARCÍA
MARTÍNEZ**

Contenido

1. Introducción	5
2. Arquitectura general del sistema	6
3. Arquitectura hardware del sistema	8
4. Arquitectura software del sistema.....	9
4.1. Bibliotecas de enlace dinámico	11
4.1.1. Pertenecientes al proyecto	11
4.1.2. Preexistentes.....	12
4.2. Programas de la solución	14
4.2.1. Programa configurador	14
4.2.2. Medidor.....	14
4.2.3. Sistema de comunicaciones	14
4.2.4. Programa visualizador	15

1. Introducción

Este proyecto está conformado por un conjunto de entidades software y hardware que se relacionan entre sí y con elementos externos.

Ha de tenerse en cuenta la existencia de múltiples componentes, tanto hardware (como por ejemplo las cámaras, el PLC y el ordenador de proceso) como software (sistema medidor, visualizador, configurador, ficheros de configuración, etc.) que han de trabajar de forma conjunta para lograr un funcionamiento correcto.

Todo esto ha de coordinarse para producir una salida que posteriormente ha de poder ser utilizada en otros sistemas.

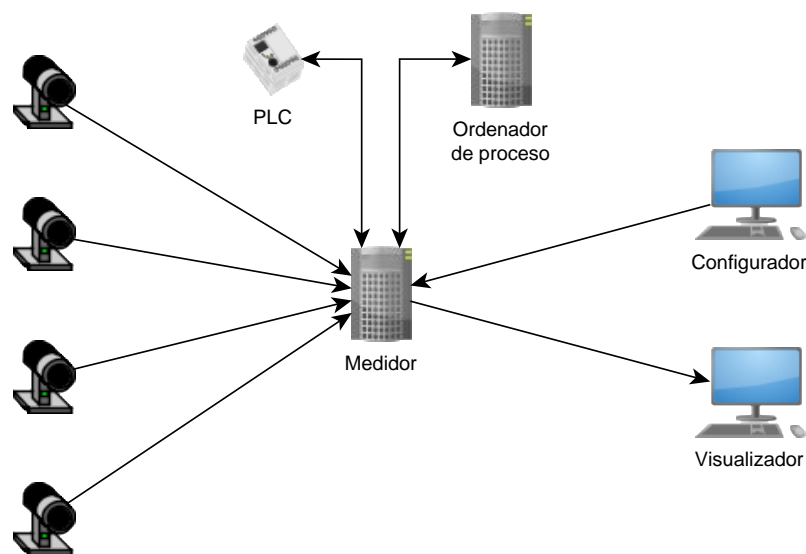


Figura 1: Diagrama simplificado del sistema

Como puede verse en la Figura 1, el sistema incluye, a grandes rasgos, los siguientes elementos:

- Un servidor de medición, o medidor, capaz de procesar las imágenes de las cámaras y extraer medidas de los perfiles que inspecciona, comunicándose también con el PLC y el ordenador de proceso.
- Un programa configurador para generar datos de perfiles y normas que luego pueda emplear el medidor.
- Un programa visualizador independiente que puede acceder remotamente a las mediciones que se estén realizando, así como a las históricas.
- Un subsistema de captura de imagen que está compuesto por cuatro cámaras, que capturan imágenes, y cuatro emisores láser que proyectan un patrón lineal sobre la superficie del carril y que se utilizan para “dibujar” su contorno.

2. Arquitectura general del sistema

El sistema implementado está formado por una serie de componentes, tanto hardware como software, así como por componentes externos que son necesarios para llevar a cabo la correcta ejecución del sistema.

De forma detallada, la estructura general del sistema puede verse en la Figura 2. A continuación, se describirá brevemente el cometido de cada módulo software implementado como parte del proyecto, así como su relación con otros elementos.

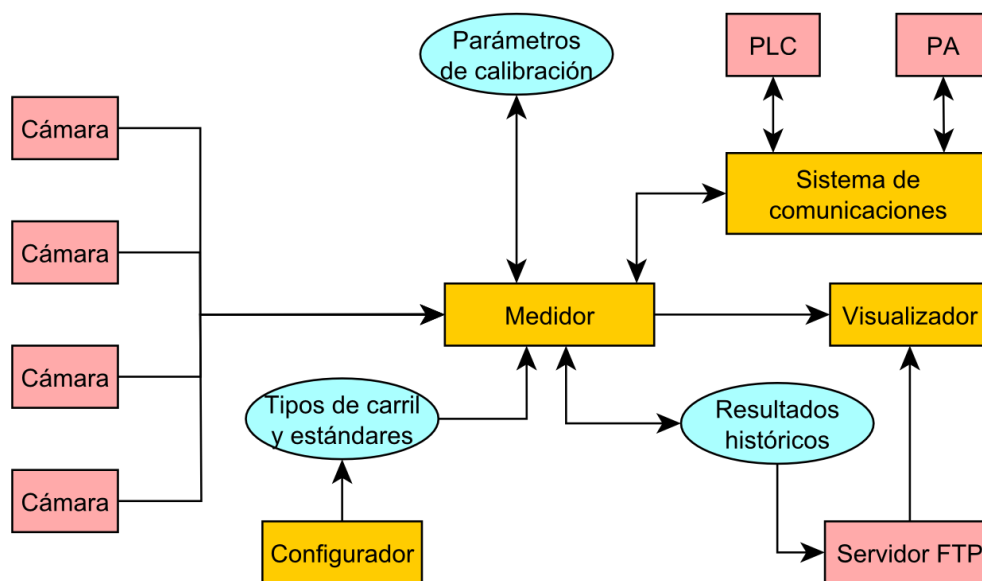


Figura 2: Estructura del sistema

En la Figura 2 se pueden observar diferentes tipos de componentes. Según su color en la figura estos son:

- En amarillo, los programas implementados como parte del presente proyecto:
 - Medidor: Lleva a cabo la medición de los carriles.
 - Sistema de comunicaciones: Comunica el programa medidor con sistemas externos.
 - Configurador: Genera un fichero de tipos de carril y normas para su uso por el medidor.
 - Visualizador: Muestra los resultados obtenidos por el medidor.
- En azul, los datos generados o consumidos por programas que forman parte del sistema.
 - Tipos de carril y estándares: Fichero que describe los tipos de carril y las normas que el programa medidor emplea.
 - Parámetros de calibración: Parámetros de las cámaras utilizadas para la medición, que el programa medidor emplea para traducir las coordenadas de los píxeles de las imágenes captadas por ellas a coordenadas del mundo.
 - Resultados históricos: Resultados de las mediciones de carriles anteriores.

- En rojo, los equipos externos o no incluidos en el proyecto, pero que tienen una estrecha relación con el mismo.
 - Cámaras: Cuatro cámaras situadas alrededor del carril que se mide y que captan imágenes para el programa medidor.
 - Servidor FTP: Servidor que proporciona los ficheros de resultados antiguos al programa visualizador.
 - PLC: Dispositivo que avisa al programa medidor de la entrada y la salida de carriles y que controla el despliegue de la plantilla de calibración.
 - PA: Ordenador de proceso, que envía al programa medidor los metadatos de los carriles que van a llegar y recibe de él los resultados de la medición.

La arquitectura software del sistema desarrollado está compuesta por varios programas junto con una serie de bibliotecas de enlace dinámico o DLL. Esta estructura se explica con mayor detalle en la sección 4.

3. Arquitectura hardware del sistema

El hardware relacionado con el sistema puede dividirse en dos tipos: el hardware específico del sistema y el hardware externo al mismo.

El hardware específico se utiliza en exclusiva para llevar a cabo las labores de este sistema:

- Cuatro cámaras, encargadas de tomar las imágenes de los perfiles cuando el *PLC* lo indique.
- Cuatro láseres, encargados de proyectar un patrón lineal que haga posible la extracción del perfil del carril.
- Una plantilla de calibración, que permite calcular la posición de las cámaras relativa al plano láser para poder interpretar correctamente las imágenes captadas.

Por otra parte, se tiene también un hardware externo, que es el encargado tanto de controlar la medición como de almacenar los resultados de las mediciones que se realicen. El hardware externo que se va a utilizar cuando el sistema se ponga en producción es el siguiente:

- Un autómata lógico programable o *PLC*. Será el encargado de controlar en qué momento se deberán tomar imágenes. Esto se realizará mediante el envío de una señal analógica (un tren de pulsos generado por un encoder) que llegará directamente a las cámaras indicando así la captura de una imagen.
El autómata tendrá también la responsabilidad de indicar al medidor la entrada o salida de los carriles al punto de inspección. Además, existe una plantilla de calibración, una placa metálica que ha de desplegarse en paralelo al plano láser para determinar la posición de las cámaras y poder compensar su desplazamiento. Puesto que esta plantilla ocupa, cuando está desplegada, un lugar por el que deben pasar los carriles, debe existir un mecanismo para desplegarla y replegarla solamente cuando sea necesario. Este mecanismo está controlado por el medidor, y el *PLC* hace las veces de intermediario, que debe ordenar el despliegue o recogida de la plantilla de calibración cuando el medidor lo solicite, tras comprobar que el sistema está en un estado adecuado (es decir, que no hay un carril presente con el que pueda colisionar la plantilla).
- Un ordenador de proceso o *PA*. Este ordenador es el encargado de controlar y supervisar el funcionamiento de la línea de producción, por lo que será el responsable de enviar la información relativa a los carriles a medir. El ordenador de proceso puede solicitar los resultados de las mediciones de cualquier carril medido, de modo que se le envíen todas las mediciones que se estén realizando.

4. Arquitectura software del sistema

Este sistema se divide en tres programas principales:

- El programa *medidor*, que es el centro de toda la arquitectura. El medidor es el encargado de tomar las medidas de los carriles mediante técnicas de procesamiento de imágenes.
- El programa *configurador*, que es el encargado de crear, editar y borrar los ficheros de configuración que el programa medidor necesite para llevar a cabo su actividad.
- El programa *visualizador*, que procesa los archivos que genere el programa medidor mostrando su contenido al usuario final.

Una secuenciación lógica para llegar al resultado final de la medición con éxito sería la siguiente:

- El programa configurador prepara el archivo de configuración necesario para la ejecución del programa medidor. Para esto no es necesario que el medidor esté en funcionamiento, ni que se encuentre en la misma máquina.
- El programa medidor captura las imágenes que se reciben de las cámaras e inicia su procesamiento. Se extraen los contornos de los carriles y se miden los valores de sus dimensiones. Estos valores se comparan con el fichero de configuración para establecer si se cumple o no con las tolerancias preestablecidas, y el resultado se muestra al usuario inmediatamente durante el proceso de medición, y se almacena además en ficheros históricos.
- El programa visualizador se conecta al medidor desde otro equipo y muestra los resultados de la medición en tiempo real. También puede obtener los ficheros históricos y mostrarlos posteriormente. Al igual que el configurador, en este último caso el visualizador actúa independientemente del programa medidor.

El motivo por el que se ha optado por esta arquitectura es que el configurador se utilizará esporádicamente y podrá ser empleado por un usuario, como puede ser un técnico de calidad, ajeno al funcionamiento del medidor. Esto mismo sucede para el caso del visualizador.

Tanto el configurador como el visualizador pueden ejecutarse en cualquier máquina de la red local, lo que hace posible que todos los programas sean usados por personas diferentes y en diferentes momentos.

Además de los tres programas comentados antes, la arquitectura software del sistema está conformada también por una serie de DLLs que se utilizarán para realizar tareas comunes a alguno de tres programas y por un programa independiente que gestiona la comunicación del medidor con elementos externos.

La Figura 3 muestra la relación existente entre los programas creados y las bibliotecas.

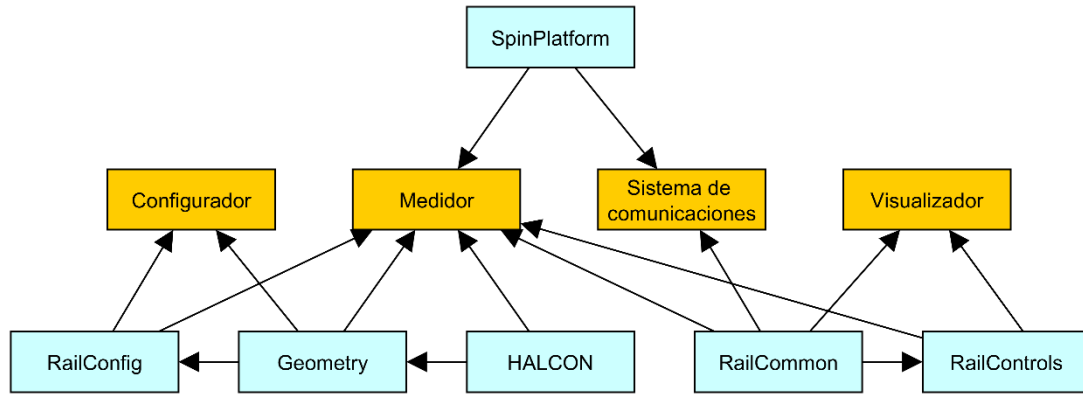


Figura 3: Relación entre programas y DLLs

En la Figura 3 se pueden observar diferentes tipos de componentes software. Dependiendo del color estos son:

- En amarillo, los programas que forman parte del proyecto.
- En azul, las bibliotecas de enlace dinámico o DLLs. Estas contienen funcionalidades comunes a varios programas del proyecto.

Tanto los programas como las bibliotecas pertenecientes al proyecto están implementados en el lenguaje de programación C#, utilizando la plataforma .NET.

4.1. Bibliotecas de enlace dinámico

4.1.1. Pertenecientes al proyecto

4.1.1.1. RailCommon

Esta biblioteca es la estructura básica del *modelo - vista - controlador* que se sigue para la visualización de los resultados. El *modelo* en esta biblioteca es la clase *Rail*, que contiene todos los resultados y los metadatos necesarios. Por otra parte está la clase *RailSource*, que actúa como controlador y que contiene una serie de métodos para exponer carriles a las diferentes vistas. Por último está la interfaz *RailView*, que actúa como *vista* y es la encargada de mostrar los resultados obtenidos, almacenados en el *modelo* y servidos por el *controlador*.

4.1.1.2. RailControls

Esta biblioteca incluye implementaciones de diferentes controles, incluyendo las vistas que implementa la interfaz *RailView*.

4.1.1.3. RailConfig

Esta DLL contiene toda la funcionalidad necesaria para generar y cargar el fichero de configuración de normas y carriles. Este fichero, generado por el programa configurador y empleado por el medidor, especifica para cada tipo de carril el contorno que debe tener idealmente (para luego compararlo con los carriles medidos) y los márgenes de tolerancia contenidos en las normas.

Esta biblioteca también es responsable de cargar y procesar los ficheros DXF que el usuario introduce en el programa configurador. Debe además almacenar, una vez leído el fichero de configuración, toda la información relativa a las normas y carriles que emplean tanto el medidor como el propio configurador.

4.1.1.4. Geometry

Esta biblioteca contiene todas las operaciones básicas de geometría que se realizan tanto en el configurador como en el medidor y en la biblioteca de configuración *RailConfig*. Esta biblioteca se utiliza para construir los perfiles mediante arcos y segmentos así como para calcular todas las dimensiones. Además, permite realizar transformaciones y todas las operaciones necesarias para llevar a cabo el proceso de medición.

La biblioteca se divide en las siguientes partes:

- *Core*: Parte central de la biblioteca, que contiene las clases de geometría básicas utilizadas en el resto de la biblioteca y en otros programas de la solución. Estas clases abarcan desde puntos o nubes de puntos hasta arcos y segmentos, así como diferentes operaciones geométricas aplicables a ellos.
- *Dimensions*: Esta parte de la biblioteca es la encargada de llevar a cabo el cálculo de las dimensiones del carril.
- *Math*: En esta parte se realizan todos los procesos que son puramente matemáticos. Estos procesos son, entre otros, la realización de aproximación de una nube de puntos a un modelo, la creación de matrices y operaciones sobre ellas, así como diferentes transformaciones geométricas.

- *Rail*: Esta es la parte encargada de crear las primitivas que componen un carril, es decir, arcos y segmentos, así como crear aproximaciones de nubes de puntos a estos segmentos e incluso crear una serie de estructuras de datos que ayuden al tratamiento de los carriles. Este tratamiento se realiza mediante una parte central llamada *RailModel*. El *RailModel* contiene toda la información relativa a los carriles, sus arcos y segmentos, así como una caché de datos sobre el propio carril, en el que se apoya el sistema de medición.
- *Registration*: Esta es la parte encargada de llevar a cabo el proceso de correspondencia entre puntos de las imágenes, los tomados mediante las cámaras y extraídos de estas, a puntos de los perfiles teóricos.

4.1.2. Preexistentes

4.1.2.1. HALCON

HALCON es una librería de visión artificial desarrollada por la empresa alemana MVTec, que incluye un entorno de desarrollo, denominado HDevelop. Además, proporciona interfaces de comunicación y drivers para numerosas cámaras de visión por computador disponibles en el mercado.

El programa medidor utiliza esta librería para calibrar el sistema de visión por computador, obtener imágenes y extraer los contornos de los carriles representados por la proyección de los patrones láser en las imágenes adquiridas. También la utiliza en menor medida la biblioteca de geometría para realizar determinadas transformaciones.

4.1.2.2. SpinPlatform

Esta parte es un conjunto de bibliotecas que conforman una plataforma desarrollada por el departamento Spin del Centro de Desarrollo Tecnológico de ArcelorMittal en Asturias.

La idea principal del uso de esta plataforma ha sido homogeneizar la estructura de todos los programas que se realicen para este departamento en concreto.

Esta plataforma contiene una serie de bibliotecas dinámicas para realizar tareas automatizadas. Cuenta con una serie de funcionalidades que abarcan desde la creación de hilos, creación y establecimiento de comunicaciones entre procesos y a través de la red, gestión y uso de sensores, etc.

Estas funcionalidades están divididas en las siguientes DLLs:

- **SpinPlatform_common**: Biblioteca principal en la que se incluye SpinDispatcher, la parte que gestiona todos los recursos e hilos de Spin, por lo que debe estar presente en todos los proyectos en los que se use esta plataforma. Aquí se incluye, además, el módulo de configuración, comunicaciones, errores y todo el conjunto de estructura de datos. También se incluye toda la gestión de hilos y comunicación entre procesos.
- **SpinPlatform_Controls**: En esta se incluyen controles genéricos implementados por Spin.
- **SpinPlatform_IO**: Aquí se encuentran los módulos de comunicación con bases de datos y de generación de consultas, un módulo para la creación de logs, así como un módulo para envío de emails y otro para comunicaciones FTP.

- **SpinPlatform_Sensors:** Integra los módulos que permiten comunicarse con sensores de instalaciones industriales.

En este proyecto se utiliza la DLL que contiene *SpinPlatform_common*, la parte central de esta plataforma, así como parte de la DLL de entrada y salida, *SpinPlatform_IO*.

4.2. Programas de la solución

4.2.1. Programa configurador

El cometido fundamental de este programa es preparar un archivo de configuración en XML, que el medidor usa después. Este fichero contiene toda la información necesaria sobre los carriles y las normas que se usan en el medidor.

Este programa se puede ejecutar de forma offline. Esto quiere decir que no se tiene por qué ejecutar a la vez que el programa medidor sino que, por el contrario, se puede utilizar para crear los archivos de una forma totalmente independiente del proceso de medición. Esto aporta una serie de ventajas, como son el crear archivos desde otro ordenador que no es el que ejecuta el programa medidor, o ser ejecutado por otras personas ajenas al proceso de medición.

Los archivos de configuración que este programa genera son complejos, puesto que contienen mucha información relativa a perfiles, normas y tolerancias, por lo que el uso de este programa es recomendable para la generación de los mismos. De todas formas se permite que un usuario avanzado pueda modificar este fichero de forma manual.

4.2.2. Medidor

El medidor constituye la parte fundamental del sistema construido en el proyecto. Implementa el subsistema de visión por computador y procesamiento de imágenes y se comunica con el resto de librerías del sistema para proporcionar las dimensiones del carril inspeccionado.

Para un funcionamiento correcto el medidor debe de estar calibrado. La calibración de un sistema de visión por computador basado en triangulación óptica mediante láser incluye parámetros intrínsecos (que permiten corregir las distorsiones ópticas del conjunto sensor más lentes) y parámetros extrínsecos (que permiten conocer la posición exacta en el espacio de cada uno de los puntos del patrón láser que observa la cámara).

Necesita conocer también los datos identificativos de los carriles que mide, proporcionados por el ordenador de proceso, y el momento en que un carril entra y sale de la zona de inspección (de la zona en la que el carril intersecta con el plano formado por los patrones lineales proyectados por los emisores láser).

También ha de tener acceso a los estándares y las definiciones de los tipos de carril que se van a medir. Esta información viene generada por el configurador.

En cuanto a los resultados, los publica de tres formas distintas: los almacena localmente, los envía a los visualizadores que se encuentren conectados al medidor y los envía al sistema de comunicaciones, para que los remita al ordenador de proceso.

4.2.3. Sistema de comunicaciones

Este sistema controla todas las comunicaciones existentes con el medidor y los agentes externos presentes, como son el *PLC* y el ordenador de proceso.

Este sistema se divide en tres grandes partes:

- La comunicación entre el medidor y el sistema de comunicaciones.

- La comunicación entre el ordenador de proceso y el sistema de comunicaciones.
- La comunicación entre el PLC y el sistema de comunicaciones.

El sistema de comunicaciones es, por tanto, el encargado de centralizar los mensajes que van desde los extremos y redirigirlos de forma adecuada al lugar de destino. Gracias a este sistema, el medidor se abstrae de las tareas de comunicaciones de red.

Para poder probar adecuadamente la comunicación con los agentes externos se ha implementado un emulador de PLC y un emulador de ordenador de proceso. Mediante estos dos emuladores se simula el comportamiento de los elementos reales que están presentes en el entorno de producción. Estos emuladores se utilizan durante la fase de desarrollo y experimentación en laboratorio y en la instalación industrial para detectar incidencias en la red o en el software de red los equipos externos que necesita el sistema.

4.2.4. Programa visualizador

Los archivos generados por el programa medidor contienen información histórica que puede resultar muy útil para varias aplicaciones en la instalación industrial, como puede ser la gestión de calidad o atención a reclamaciones de clientes sobre los carriles fabricados.

Por lo tanto, el programa visualizador puede llevar a cabo un análisis histórico de la producción de una forma totalmente independiente del proceso de medición, lo que implica que no se tiene que realizar este análisis en el mismo ordenador donde se están tomando los datos, sino que se pueden compartir, mediante una red local por ejemplo, los archivos que contengan la información histórica y realizar el análisis en cualquier otro lugar, como lo es por ejemplo el despacho de los ingenieros de calidad.

Por lo tanto, como se ha explicado anteriormente, con la finalidad de poder visualizar y analizar esta información desde cualquier PC, se incluye en el proyecto un programa visualizador más completo que el de tiempo real que ejecuta el medidor, y que es a su vez totalmente independiente de este.