

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA  
INDUSTRIAL**

**DEPARTAMENTO  
de Ingeniería de Sistemas y Automática**

**ÁREA  
Ingeniería de Sistemas y Automática**

# **CONTROL EN POSICIÓN DE UN EJE DEL ROBOT SCORBOT-ER 4PC**

---

**Autor: Enrique Arrabal Almagro**

**Director: Víctor Torres López**

**Titulación: Graduado en Ingeniería en Tecnologías Industriales**

**MÁLAGA, septiembre de 2015**

# Índice General

1	Glosario	6
2	Introducción	6
3	Objetivos	7
4	Estructura de la memoria	7

[scale=0.25]./1-Portada/img/uma  
Portada/img/etsii

[scale= 0.25]./1-

## List of Figures

## Lista de Tablas

# 1 Glosario

- API
- frontend
- backend
- PID

## 2 Introducción

Este trabajo se concibe con el objetivo de evaluar Golang Go como lenguaje de programación y entender los aspectos positivos y negativos a la hora de usarlo en un entorno profesional. Para ello se necesita abordar un proyecto con suficiente entidad, permitiendo a llevar lo más al límite posible el lenguaje, y a la vez que cuya complejidad no haga que el trabajo se escape del contexto y alcance con el que están concebidos. Vamos a centrarnos en poner a prueba puntos que el lenguaje vende como bondades o que forman una parte en la mayoría de soluciones:

- comunicación entre sistemas con distintos protocolos
- asincronía y gestión de concurrencia
- ecosistema, documentación y comunidad del lenguaje
- sintaxis, facilidad de mantener la simpleza y limpieza en el código

para probar todos estos conceptos se concibe el siguiente sistema:

- Un servicio central de control
- una interfaz de usuario para controlar de forma amigable el sistema central
- un software cliente que recepcione comandos creados en el sistema central
- un programa de control automático que ejecute dicho comando

la comunicación entre sistemas será por RPC para poner a prueba con un sistema de comunicación que tiene paquetes experimentales y en desarrollo, esto nos ayudará a ver el ecosistema a la hora de buscar documentación, paquetes, soluciones estandarizadas

en el sistema central de control se pondrá a prueba la facilidad para implementar arquitecturas más estandar para este tipo de sistemas como son las API crud para la gestión de datos y a su vez luego la comunicación con los sistemas clientes. Es donde se centrará el grueso del tiempo del proyecto ya que en un software profesional la arquitectura hexagonal en estos tipos de sistemas es un desarrollo básico y ampliamente extendido para poner a prueba los conceptos generales. con ello también aprenderemos a manejarnos con lo básico que ofrece el lenguaje de programación y así enfrentar detalles de implementación más específicos como son la asincronía o la algoritmia de control PID con más soltura.

en el frontal podremos poner a prueba la facilidad de maquetación e iteración en el diseño

en el software cliente pondremos a prueba una de las características más importantes del lenguaje y es su capacidad para ser compilado en distintas plataformas sin máquina virtual, es un lenguaje compilado y el resultado es un ejecutable nativo en la máquina final. aquí se plantea ser compilados en arquitecturas típicas de pc personal como son los intel o amd y luego la maquina final sera en una raspberry

programa de control automático, se pondrá a prueba la versatilidad del lenguaje para desarrollar algoritmos de la forma más limpia y mantenible posible.

como resultado se tendrá un sistema que contendrá los elementos más típicos de una solución comercial hoy en día. un sistema web API y servicios intercomunicandose a través del mismo, Un frontal y servicios clientes.

Se va hacer incapié en el proceso y no solo en el resultado, aplicando todos los procesos que se siguen como referencia en la industria a día de hoy. Las principales herramientas o conceptos que a día de hoy se usan se puede categorizar de la siguiente forma:

- respecto a cómo se va entregando valor de forma iterativa: scrum, kanban, agile, extreme programming...
- respecto a cómo se garantiza cumplir la expectativa del cliente con el producto la calidad y la robustez del mismo: Testing (TDD,BDD...) diseño de casos de uso, diagramas UML, UX, UI
- respecto a cómo se garantiza la posibilidad de cambio y mantenimiento: arquitectura de capas o hexagonal, DDD...

De todas ellas, hemos escogido como relevantes para este trabajo las herramientas de diseño que garantizan la posibilidad de cambio en el futuro y el cumplimiento de las expectativas del cliente: arquitectura de capas, DDD y testing. aunque se utilicen todas no se hará incapié en ellas en este documento.

### 3 Objetivos

Crear un sistema capaz de albergar y gestionar tareas para su ejecución en remoto.

Para ellos se requerira un servicio web API (application programming interface) y dos servicios clientes uno para manejarlo el frontend y otro para ejecutar las tareas

Para hacer uso en mayor medidas de la herramienta escogida y para explorar su uso en nuestro campo la tarea a ejecutar en remoto sera un control PID en velocidad y posición de un motor de corriente continua.

Tendremos la capacidad de:

- Añadir editar, eliminar y obtener las tareas contenidas en el sistema
- Añadir editar, eliminar y obtener los resultados de la ejecución de dichas tareas
- Añadir editar, eliminar y obtener las direcciones de los equipos remotos donde se ejecutarán las tareas

El servicio cliente remoto tendrá la capacidad de recibir un comando, ejecutarlo y devolver el resultado.

## 4 Estructura de la memoria

La estructura del presente documento va a perseguir mostrar el proceso como si de un proyecto real se tratara. Se va a comenzar presentado las herramientas que se pretende utilizar. Luego un dise;o a modo de proyecto tal y como se le presentaria a un cliente para que aprobara su comienzo de ejecucion. Se va a proceder a describir los hitos importantes en el desarrollo del mismo y las adaptaciones que surgen debido a imprevistos y descubrimientos de la propia investigacion contemplada en el proyecto

Proyecto básico: se va a diseñar el sistema objetivo tal para su presentación y aprobación por cliente. Está compuesto por una memoria descriptiva, constructiva y económica Ejecución del proyecto: se ha documentado el desarrollo del proyecto con sus distintos hitos y modificaciones requeridas por el devenir de la ejecución Aplicaciones del sistema: pruebas y resultados. Latencias! Control de luces Control del motor Conclusiones: se expondrán las distintas conclusiones que se han extraido con respecto a los puntos objetivo. Golang, protocolo de comunicación, diseño de software y testing Lineas Futuras