Universidad San Jorge

Escuela de Arquitectura y Tecnología

Grado en Ingeniería Informática

**Proyecto Final**

**Programación de la lógica de control de un brazo robótico**

**Autor del proyecto: Vicente Francisco Lozano Vicente**

**Director del proyecto: Carlos Bello Gimeno**

**Zaragoza, 13 de junio de 2019**

Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado en Ingeniería Informática por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título.

Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido.

Doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Firma Fecha 13/06/2019

**Dedicatoria y Agradecimiento**

Tabla de contenido

[Resumen 1](#_Toc13418846)

[Abstract 1](#_Toc13418847)

[1. Introducción 3](#_Toc13418848)

[2. Estado del Arte 4](#_Toc13418849)

[2.1. Un poco de historia 4](#_Toc13418850)

[2.2. Aplicaciones para control de uArm disponibles 4](#_Toc13418851)

[2.3. Otros tipos de controladores 4](#_Toc13418852)

[2.3.1. LeapMotion 4](#_Toc13418853)

[2.3.2. OpenMV 4](#_Toc13418854)

[2.3.3. Smartphone 4](#_Toc13418855)

[2.3.4. Raton y Teclado 4](#_Toc13418856)

[2.3.5. Otros 4](#_Toc13418857)

[3. Objetivos 6](#_Toc13418858)

[3.1. Objetivos iniciales del proyecto 6](#_Toc13418859)

[3.2. Objetivos de la empresa 6](#_Toc13418860)

[3.3. Como implementamos los objetivos 6](#_Toc13418861)

[4. Metodología 7](#_Toc13418862)

[4.1. Comparativa de las diferentes metodologías y eleccion 7](#_Toc13418863)

[4.2. Particularidades de la adaptación de la metodología 7](#_Toc13418864)

[4.3. Detalles generales 7](#_Toc13418865)

[4.4. Evolucion de los Sprints 7](#_Toc13418866)

[5. Estudio económico 8](#_Toc13418867)

[5.1. Coste material del proyecto 8](#_Toc13418868)

[5.2. Coste en Horas del desarrollo 8](#_Toc13418869)

[5.3. Estimaciones de rentabilidad 8](#_Toc13418870)

[6. Resultados 9](#_Toc13418871)

[7. Conclusiones 10](#_Toc13418872)

[8. Bibliografía 11](#_Toc13418873)

[9. Anexos 12](#_Toc13418874)

# Resumen

En este proyecto estudiamos como controlar un brazo robótico a través de comandos, ponemos a prueba los distintos rangos operacionales del mismo y lo conectamos a distintas tecnologías para dotarlo de sensores e inteligencia para resolver tareas. El brazo en cuestión es un uArm Swift Pro y la cámara una OpenMV.

# Abstract

In this Project we study the different modes of control of a robotic arm using serial commands, test the capabilities of the arm and connect it to different parts to give it sensing capabilities and intelligence to react to the input and solve tasks. The robotic arm model is uArm Swift Pro and the camera is an OpenMV.

# Introducción

Un brazo robótico es una herramienta multifunción de propósito general que se puede adaptar a casi cualquier tarea. El precio de esta adaptabilidad es una menor eficiencia para una tarea concreta y repetitiva y la mayor complejidad en la programación de la lógica del mismo.

Están compuestos por 2 partes diferenciadas y una opcional, a saber, el cuerpo del brazo, los actuadores y los sensores.

El cuerpo queda definido por sus medidas, sus grados de libertad y sus pesos máximos.

El actuador puede ser de varios tipos, y las diferencias entre varios son especificas de que tipo de actuador estemos hablando, en caso de que sea un actuador destinado a la sujeción y movimiento de piezas, queda definido por el peso máximo y las condiciones de sujeción.

Respecto a los sensores, hay una variedad sin fin, ya que cualquier sensor de propósito general, puede ser adaptado para trabajar con un brazo si sus interfaces son compatibles.

En este proyecto trabajamos con un uArm Swift Pro, que es un aparato bastante completo para labores de prototipado e investigación. No es un brazo funcional para producción ya que sus pesos máximos no están en rangos relevantes. No obstante, todo lo realizado aquí seria extrapolable a cualquier otro modelo con mínimo esfuerzo.

Un brazo robótico por si solo simplemente es capaz de mover un actuador a una posición dentro de su rango operacional, para enriquecer esta tarea y que sea capaz de resolver problemas mas complejos, hay que dotarlo de capacidad sensora para poder evaluar su entorno y reaccionar correctamente. Incluso dentro de los entornos de fabricación, donde todo movimiento es repetido milimétricamente y sin variación, que el brazo sea capaz de detener su funcionamiento si un operario esta en peligro, o alterar ligeramente su funcionamiento si una pieza aparece desplazada o erróneamente posicionada en la línea puede ayudar a ahorrar costes y eliminar riesgos.

Para desarrollar un producto generalista, el sensor mas interesante es el óptico ya que los seres humanos utilizamos gran cantidad de códigos visuales y ser capaz de interpretarlos le permitirá interactuar en nuestro mundo. Por ello, como principal sensor, vamos a trabajar con una cámara modelo OpenMV que posee suficiente resolución y autonomía para nuestras necesidades.

Equiparemos esta cámara con un software de visión por computador para distintas tareas y analizando los estados en los que nos encontremos tomaremos las decisiones de movimiento que permitan o bien solucionar el problema o bien adquirir la información necesaria para ello.

# Estado del Arte

## Un poco de historia

La tecnología de los brazos robóticos no es nueva, Leonardo Da Vinci, ya en 1495, construyo el primer autómata conocido con brazos. Mas adelante, con el advenimiento de los circuitos integrados, Willard Pollard patenta en 1938 el primer brazo robótico moderno cuya tarea era pintar con un spray, aunque nunca llego a construirlo.

Fue ya en 1962 cuando se dio el paso a la industria y se instauro en una fabrica de GM el primer brazo robótico en producción, también encargado de tareas de pintado, del cual se utilizaron 8500 unidades.

Desde entonces, han tenido una presencia ubicua en la industria de fabricación, y han ido mejorando tanto en grados de libertad y alcance como en repetibilidad y precisión.

Ahora mismo, se esta trabajando en varios campos, mejorando la precisión del brazo, reduciendo su tamaño o añadiendo capacidades sensitivas para extender su utilidad a campos para los que todavía no esta maduro o en los que hay margen de mejora como en aplicaciones sanitarias.

Un robot cirujano por poner un ejemplo, necesitaría de retroalimentación háptica, para que el operador pudiera sentir la presión ejercida y un nivel de precisión milimétrico. Si el brazo tiene inteligencia, además, necesitaría muchos más sensores y lógica que hoy en día está fuera de nuestro alcance.

Nosotros vamos a trabajar en la interconexión de sensores con el brazo y la lógica que maneja esa conexión para testear y demostrar las capacidades del mismo y buscarle aplicaciones nuevas.

## Aplicaciones para control de uArm disponibles

La única aplicación disponible para control y demostraciones de uArm es el propio uArm Studio. Esta aplicación permite controlar el uArm a través de una UI que usa el ratón y el teclado, así como enviar secuencias de comandos usando la herramienta de scripting, que utiliza el lenguaje Blocky. Estos scripts residen en el ordenador y **(no?)** pueden ser cargados en la memoria del uArm, requiriendo, por tanto la conexión de un PC al mismo.

## Otros tipos de controladores

### LeapMotion

Sistema de cámaras infrarrojas y un conjunto de librerías para trabajar con las mismas que permite la captura de manos, dedos y otros artefactos en tiempo real, siempre y cuando se encuentren en el POV de las cámaras.

### OpenMV

Cámara de visión por computador muy sencilla, con su propia memoria y una implementación de OpenCV reducida (OpenMV), ideal para pequeñas aplicaciones de bajo consumo y autónomas. Se conecta al brazo y envía comandos al mismo por serie dentro de las capacidades de la propia cámara.

### Smartphone

Cualquier Smartphone dispone de una pantalla táctil y gran número de sensores inerciales y de otros tipos. La omnipresencia de estos aparatos, unido a su interconectividad excelente nos permiten usarlos como dispositivo de entrada para el control del brazo, a través de conexión directa, emparejamiento por Bluetooth o conexión por datos a un servidor. Separando de esta forma el controlador del actuador en el espacio, abriendo la posibilidad a nuevas aplicaciones. Esto toma mayor importancia con la implantación de 5g.

### Ratón y Teclado

Periféricos clásicos, pero no por ello menos útiles. Gran parte de la población está habituada a estos periféricos y tienen gran precisión.

### Otros

Joysticks, Mandos y otra gran variedad de periféricos pueden ser interesantes para el futuro o aplicaciones imprevistas en el momento de entrega de este proyecto, pero si el modulo existe y funciona, siempre se dispone de la posibilidad de utilizarlo. Además, actualmente, el único actuador del que disponemos es el brazo robótico uArm, pero, debido a la estructura del ecosistema software, el añadido de un actuador distinto podría hacerlos relevantes.

# Objetivos

## Objetivos iniciales del proyecto

Los objetivos que se buscan con la realización de este proyecto son los siguientes:

1. Introducir a el/los alumno/alumnos en los conceptos básicos de sistemas de control y robótica.
2. Introducir a el/los alumno/alumnos en los conceptos básicos de sistemas de visión por computador.
3. Obtener experiencia en el desarrollo de un sistema de control para un caso concreto de brazo robótico fácilmente extrapolable a un entorno industrial real.
4. Adquirir un conocimiento extenso de los dispositivos LeapMotion y OpenMV, así como de la tecnología en que se fundamentan y sus capacidades para el desarrollo.
5. Saber realizar cuadros de mando dentro de un sistema de control.

## Objetivos de la empresa

Los objetivos de la empresa son:

* Tener una UI única para controlar el brazo con diferentes periféricos
* Esta UI debe de poder al menos, dar control sobre el brazo de las maneras que la suite actual permite.
* Deberá poderse controlar el brazo con el LeapMotion
* Deberá poderse controlar el brazo con scripts en la cámara OpenMV
* Se valorara el añadido de nuevas formas de control del brazo
* Se valorara la facilidad de agregación de nuevos periféricos
* Se valorara la realización de demos y ejemplos para facilitar la integración en el Showroom.

## Como implementamos los objetivos

Los puntos 1,3 se cumplen al realizar el control por comandos del brazo usando su propio sistema de comunicación serie

El punto 5 se cumple en el desarrollo de la UI de control y operación

Los puntos 2 y 4 se cumplen realizando los módulos de control mediante OpenMV y LeapMotion.

# Metodología

Una de las primeras fases en la ejecución de un proyecto software, o de cualquier otro tipo, es la elección de la metodología que más se adapte al proyecto y al equipo de trabajo.

Hoy en día, debido a la alta demanda de retroalimentación y lo complejo que puede llegar a resultar tener una comunicación fluida y eficiente dentro de un espacio de posibilidades prácticamente infinito como el que brindan los desarrollos software, suelen adoptarse metodologías agiles, que, por su naturaleza, brindan la posibilidad de realizar comprobaciones y correcciones de los requisitos con mucha rapidez y adaptar la prioridad de los módulos desarrollados conforme las necesidades cambiantes del cliente, sujeto a la variabilidad del mercado, varían.

## Comparativa de las diferentes metodologías y elección

Para decidir que metodología vamos a aplicar, lo primero es ver que características tiene nuestro proyecto.

* El grado de incertidumbre de las tareas es elevado, ya que, son tecnologías nuevas con las que el equipo de desarrollo no ha trabajado previamente.
* La prioridad de cada una de los requisitos es muy variable e incierta, si la siguiente presentación es un entorno u otro, puede ser necesario priorizar algo que sea relevante para ese Showroom.
* El proyecto abarca más tiempo del que hay disponible, con lo que no se da por terminado el mismo al cumplir una serie de funcionalidades, sino que el límite es el tiempo del que se dispone de la presencia del equipo de desarrollo.

Todo esto hace que descartemos metodologías más clásicas, como podría ser cascada, y optemos por una metodología ágil, ya que, nos permite:

* Entregas cortas de productos viables y usables
* Corrección de la prioridad de las funcionalidades deseadas
* Corrección de las estimaciones de tiempo
* Desarrollar el mayor numero de funcionalidades posible en el tiempo asignado para aportar el mayor valor a cliente.

De entre las metodologías Agiles, se elige SCRUM, el motivo de la elección de la misma es que, tanto el equipo de desarrollo, como el cliente ya han trabajado con ella anteriormente y cumple con los requisitos de inmediatez, adaptabilidad y entrega temprana.

## Particularidades de la adaptación de la metodología

Vamos a utilizar SCRUM clásico, realizaremos sprint de 2 semanas de duración y al final del mismo se entregara un producto viable que podrá o no entrar en producción a decisión de cliente. Cuando hablemos de cliente, estaremos hablando del departamento de I+D+i de Inycom, el cual representara Carlos Bello en el Rol de “Product Owner”.

El equipo de desarrollo está formado por 1 persona, Vicente Lozano, el alumno que también toma el rol de “SCRUM Master”.

El producto viable incluye la instalación y configuración del equipo sobre el que se realizaran las pruebas y no entra en el alcance del proyecto la configuración en otros equipos o la compatibilidad con los mismos, si bien puede haber excepciones a esto a petición de cliente.

El producto viable incluye también la memoria del proyecto y la documentación del software así como las demostraciones del mismo. Por tanto, también se incluye en el “Backlog” las diferentes partes de la memoria y la elaboración de los PowerPoint y reserva de tiempos para las demostraciones.

## Evolución de los Sprint

# Estudio económico

## Coste material del proyecto

El brazo en particular tiene un coste de USD, la cámara 43,60 Euros y el LeapMotion 70,30 Euros (11/03/2019). En su total los materiales necesarios ascienden a 900 euros.

## Coste en Horas del desarrollo

El proyecto dispone de **XXXX** horas de desarrollo, las cuales están cuantificadas en las distintas tareas.

## Estimaciones de rentabilidad

En este proyecto la rentabilidad es muy difícil de calcular, los Showroom son demostraciones técnicas que se realizan con diferentes finalidades, principalmente, captación de clientes y captación de trabajadores. Si después de una Showroom hay un nuevo contrato es complicado asignar el motivo del mismo, no obstante para estimarlo se utilizara un sistema de encuesta tras la realización de los Showroom en los que, de forma anónima se recopilaran datos sobre lo que más ha impactado en el caso de alumnos o lo que le ha parecido más interesante a un cliente potencial. También de forma indirecta en conversaciones o analizando lo que se pide, se añadirá una valoración del Showrunner y con estos datos se puede realizar una estimación.

# Resultados

Tras terminar el desarrollo del proyecto, podemos asegurar que se han cumplido los objetivos iniciales planteados. El proyecto ha sufrido varios retrasos técnicos y cambios de prioridad debido a diferentes situaciones externas, pero debido a la metodología elegida y la gestión de riesgos, se ha podido subsanar y llegar a la fecha de entrega cumpliendo las expectativas de cliente.

El producto que se entrega consta de una interfaz de usuario principal sobre la que se cumplen los diversos casos de uso solicitados por el cliente, permitiendo a un usuario sin experiencia controlar el brazo robótico uArm a través de varios periféricos de control:

* Ratón
* Joystick
* LeapMotion
* Smartphone

También se pueden cargar desde el UI diferentes scripts en la memoria de la cámara OpenMV para que esta controle al brazo en función de lo que vea la cámara, estas demostraciones incluyen:

* Resolución de puzle simple
* Tracking de objeto

Los diferentes periféricos de control, pueden ser utilizados por conexión directa o por conexión remota.

Todo esto permite realizar demostraciones en diferentes ámbitos:

* Operación en entornos estancos
* Tele operación
* Control natural integrado sin contacto
* Gamificación
* Operación Autónoma

Así, el Showrunner puede adaptar el Showroom a las necesidades del cliente con el que va a trabajar y tener significancia en la semántica de negocio del cliente final.

Todo esto, está realizado en un ecosistema modular que permite el añadido de nuevos periféricos o la sustitución del actuador (uArm en este caso) con un desarrollo mínimo, permitiendo la interoperabilidad y facilitando el mantenimiento de la suite.

# Conclusiones

El proyecto ha sido muy interesante, aunque he tenido que aprender muchas tecnologías diferentes para poder desarrollar cada uno de los módulos. Quedan varias propuestas por desarrollar que han surgido durante el desarrollo y que podrían generar más valor y podrán ser afrontadas por otro equipo de desarrollo o el equipo residente de cliente, como por ejemplo la conexión con unas gafas 3D para mejor operación en remoto del brazo, el añadido de un sistema de movimiento para el mismo con altura regulable para poder desplazar el actuador por las instalaciones, etc.

# Bibliografía

# Anexos

