



iDRAW

Interactive Drawing Robot
for Advanced Work

Statement of Work

Contenido

| | |
|---|----|
| Introducción..... | 3 |
| Integrantes del proyecto..... | 6 |
| Alcance | 7 |
| Objetivo del proyecto | 7 |
| Requisitos del sistema | 7 |
| Restricciones del proyecto | 8 |
| WBS General | 8 |
| Subsistema de diseño de la aplicación: WBS y desarrollo | 9 |
| Subsistema de control: WBS y desarrollo | 15 |
| Periodo y planificación..... | 18 |
| Periodo de trabajo | 18 |
| Diagrama de Gantt..... | 18 |
| Análisis de riesgos..... | 19 |
| Calendario de entregables..... | 22 |
| Criterios de aceptación | 23 |
| Criterios de aceptabilidad | 23 |
| Matriz de verificación | 23 |
| Plan de pruebas | 25 |
| Material y presupuesto | 26 |
| Material necesario..... | 26 |
| Presupuesto | 26 |
| Referencias | 27 |
| Figura 1: Esquema iDRAW | 27 |
| Figura 2: e-David | 27 |
| Conclusión | 28 |

Introducción

La robótica cumple un papel fundamental en el panorama tecnológico actual. Su asombroso avance en las últimas décadas le ha permitido conquistar el sector industrial, llegando a sustituir la intervención manual en aquellos procesos más rutinarios de un sistema de producción, provocando un aumento en la productividad, así como en la flexibilidad y seguridad con las que se llevan a cabo las tareas de producción.

Sin embargo, el avance de este campo no se detiene aquí. Fuera del entorno industrial, no son menos sus aplicaciones, como son el caso del sector espacial o militar. Para más inri, en los últimos años se han comenzado a desarrollar aplicaciones enfocadas al sector servicios, tales como la medicina, la enseñanza, la seguridad o incluso el hogar. El futuro de esta disciplina apunta en dirección a una mejora en la autonomía y destreza con las que el robot ejecuta su función, para facilitar su aplicación a todos los casos posibles.

Un gran punto a favor del uso de robots es la flexibilidad que presentan a la hora de realizar un cierto grupo de funciones, ya que estos pueden ser reprogramados para llevar a cabo diferentes tareas, ahorrando en costes y en tiempo de aprendizaje. Sin embargo, se dan casos en los que la “pre-programación” no es posible o, al menos, no recomendable. Se trata de situaciones en las que la labor a llevar a cabo tiene lugar en localizaciones de difícil acceso, o que conlleven un alto riesgo para el operario, y en las que la serie de órdenes a seguir por el robot son parcial o totalmente desconocidas a priori, haciendo indispensable una supervisión humana. Imagínese el caso de un pequeño robot manipulador cuyo efector final sea un soldador, encargado del mantenimiento de una cierta estructura con localizaciones inaccesibles para una persona; ante la imposibilidad de saber las características de la incidencia a solventar, será un operario quién, en función de la información aportada por el robot, establecerá la serie de pasos a efectuar por el mismo para completar la labor de mantenimiento.

Por tanto, resulta de interés la idea de un robot capaz de responder a órdenes en tiempo real, de manera que no sea necesario que estas estén ya programadas. Con todo esto, el objetivo de este proyecto será el desarrollo de una aplicación capaz de transmitir órdenes de movimiento basadas en trazos realizados sobre una pantalla y que el robot manipulador las interprete en tiempo real con relativa fluidez y precisión, lo que permitiría ampliar la destreza de movimiento respecto a un robot programado; así como también la puesta en marcha del brazo robótico en cuestión. En este caso, se tratará con un manipulador cuyo efector final será un utensilio de escritura, de manera que su función sea trazar sobre un papel el mismo dibujo que se está pintando sobre la pantalla.



Figura 1: esquema del iDRAW

Algo similar es lo que consigue el robot **e-David** (*"Drawing Apparatus for Vivid Interactive Display"*) desarrollado por la Universidad de Konstanz, capaz de representar una imagen previamente adquirida por cámara. El siguiente paso en este desarrollo artístico sería la creación propia de imágenes por parte del robot, en lugar de copiarlas una vez captadas; y casos como estos son los que pueden encontrarse en la competición *Roboart*, donde robots como **Cloudpainter**, ganador de la edición de 2018, utilizan la Inteligencia Artificial para crear obras de arte.

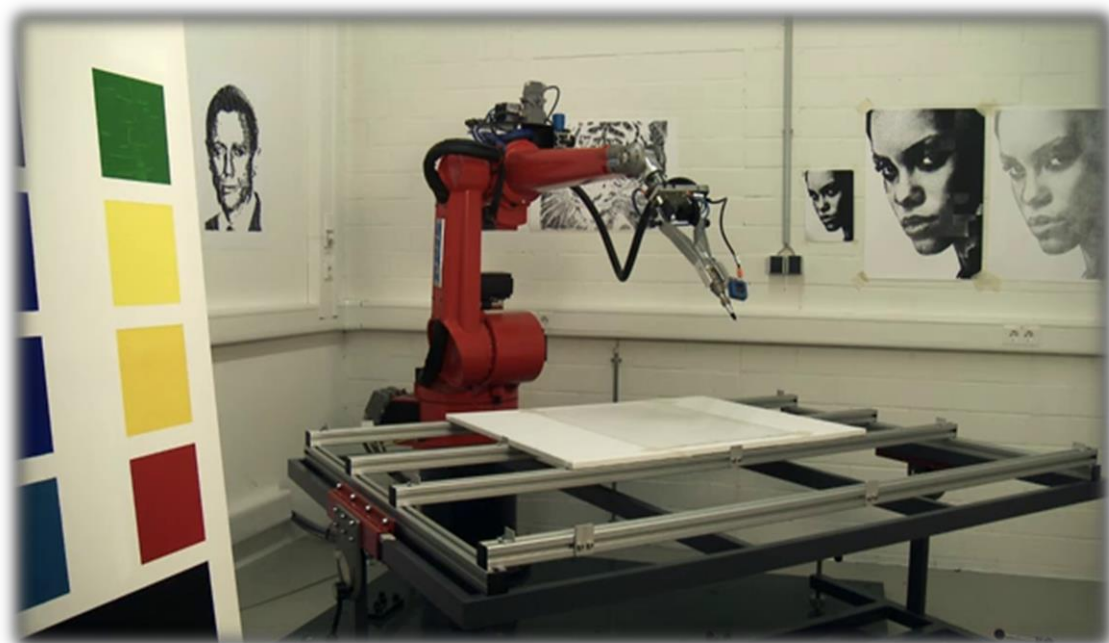


Figura 2: e-David

Es importante recalcar que, si bien la finalidad de los robots expuestos en el párrafo anterior puede parecer al misma que la que se pretende, en el caso de este proyecto se quiere llevar a cabo la representación del dibujo en tiempo real, de manera que robot y operario dibujen más o menos simultáneamente, lo cual se traduce básicamente en la captación de trazos y su correspondiente transformación a movimiento en el menor tiempo y con la mayor precisión posibles. Es, por tanto, que este objetivo se aproxima más al paradigma de la telerrobótica.

El uso de robots teleoperados es bastante extendido, dadas las razones expuestas anteriormente. Al lector le será familiar el uso de esta rama de la robótica en exploraciones espaciales, o incluso, quizás menos familiar por tratarse de una aplicación más reciente, la implementación de esta tecnología en entornos de operación quirúrgica. De hecho, son grandes los avances conseguidos en relación al control de movimiento por gestos mediante la tecnología Kinect o similares, logro que se aproxima bastante a lo que aquí se pretende; aunque este caso se centra en la interpretación de gestos con carácter bidimensional, es decir, trazos sobre una superficie plana.

En la manera en que se ha descrito todo lo anterior, este resultado no quedaría restringido únicamente al arte de la pintura. Con la correspondiente modificación de la herramienta a utilizar y el correcto acondicionamiento del manipulador al entorno de trabajo, son numerosas sus posibles aplicaciones, tales como el ejemplo del soldador descrito anteriormente.

Integrantes del proyecto

Este proyecto tiene 8 integrantes que van a realizarlo:

- **PRESIDENTE Y AYUDANTE:** Pedro José Díaz García y David Rodríguez Fernández.
 - Se encargarán de la elaboración del Statement of Work (SOW), exceptuando las secciones correspondientes a los riesgos y el presupuesto, que serán realizadas por el técnico de riesgo y maqueta.
 - Deberán coordinar al resto del equipo, asegurándose de que se cumplan plazos y objetivos. Al mismo tiempo, colaborarán con estos en las restantes tareas como medio de apoyo, a la par que consiguen una mejor percepción de las labores que se están realizando con el fin de una correcta elaboración del SOW.
- **CONTROL Y MARKETING:** Álvaro Sánchez Cano y Javier Gómez Jiménez.
 - Se encargarán, en primer lugar, de la adquisición y montaje del brazo robótico empleado en el proyecto. Una vez hecho esto, desarrollarán un controlador basado en código C para manipular correctamente el movimiento del brazo robótico.
 - Por último, realizarán grabaciones del robot en funcionamiento que serán utilizados con fines de marketing.
- **APLICACIÓN JAVA:** Antonio Cejas Laurel, Guillermo García Domínguez y Raúl Olmo Ruiz.
 - Se encargarán del desarrollo de la aplicación, basada en lenguaje Java, utilizada para transmitir los movimientos solicitados al brazo robótico. Esto incluirá el diseño de una interfaz y la configuración de la comunicación app-robot mediante tecnología Bluetooth.
- **TÉCNICO EN RIESGOS Y MAQUETACIÓN:** Rubén López Rey.
 - Se encargará de redactar las secciones del SOW correspondientes a los materiales empleados y los riesgos que conlleva el proyecto, con el pertinente análisis de riesgos. Como parte de dicho análisis, llevará a cabo el diseño gráfico, de un brazo robótico similar al empleado, que será utilizado como plan de contingencia.

Alcance

Objetivo del proyecto

El objetivo principal del proyecto es implementar la capacidad de representar en un A4 en tiempo real, mediante un brazo robótico, aquello que introduzca el usuario en la app de su terminal. Este objetivo se detalla más en los requisitos.

Requisitos del sistema

Se han dividido los distintos requisitos según su utilidad:

Requisitos funcionales

- F.1. El brazo robótico se moverá fluidamente en su entorno de trabajo.
- F.2. El brazo robótico será capaz de pintar puntos y trazos tanto continuos como discontinuos recibidos mediante un pad a través de una aplicación diseñada en Java.
- F.3. El sistema podría trabajar en tiempo real.
- F.4. El sistema podría ser sensible a la velocidad de escritura.

Requisitos de prestaciones

- P.1. El brazo escribirá sobre el papel con un retardo máximo de 10 segundos con respecto a la aplicación.
- P.2. El brazo recorrerá posiciones en su espacio de trabajo en menos de 15 segundos.

Requisitos de diseño

- D.1. El brazo robótico debe tener un rango de operación de aproximadamente 40x40x40 cm; para pintar de manera fluida sobre un folio tamaño A4.
- D.2. El brazo robótico debe tener un diseño que facilite la reparación y sustitución de cualquier componente.
- D.3. La aplicación debe tener un diseño atractivo, pero priorizando su correcta funcionalidad.
- D.4. La aplicación debe tener diferentes modos de funcionamiento (modo botones, modo pad, etc.)
- D.5. El brazo robótico debe tener una garra que garantice una correcta sujeción del rotulador.

Requisitos de operación

- O.1. El brazo robótico se controlará de forma remota desde un dispositivo móvil mediante Bluetooth.
- O.2. La aplicación debe estar disponible en diferentes dispositivos móviles.
- O.3. La aplicación estará disponible en la nube mediante un código QR para un uso libre y gratuito.

- O.4. El brazo debe estar operativo para pintar en menos de 10 segundos y ser capaz de calibrarse en menos de 5 segundos.
- O.5. La aplicación debe estar operativa en menos de 5 segundos y ser capaz de reiniciarse en menos de 10 segundos.

Requisitos de software

- SW.1. La aplicación de Java debe captar con gran precisión los trazos a realizar por el brazo robótico.
- SW.2. El control del brazo robótico debe ser rápido para garantizar un tiempo de retardo bajo, pero a su vez fiable y seguro para representar con exactitud los trazos sobre el papel.
- SW.3. La transmisión Bluetooth debe ser rápida y segura para garantizar la correcta transmisión de datos desde la aplicación al brazo robótico.

Requisitos mecánicos

- M.1. Los motores del brazo tendrán una velocidad de reacción mediana-alta para operar con el mínimo retraso posible.

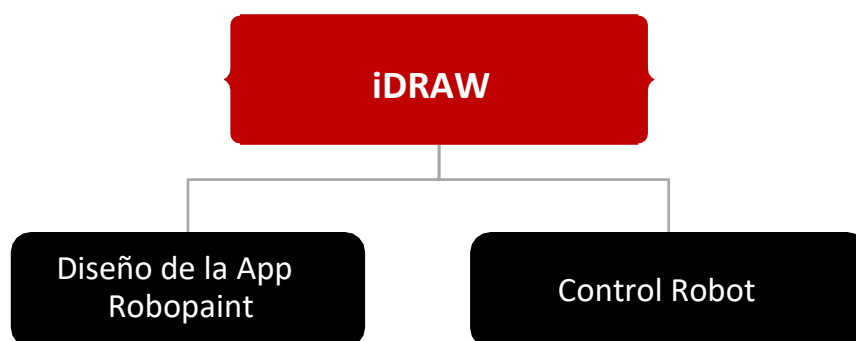
Restricciones del proyecto

Las restricciones expresadas por el cliente se redactan a continuación:

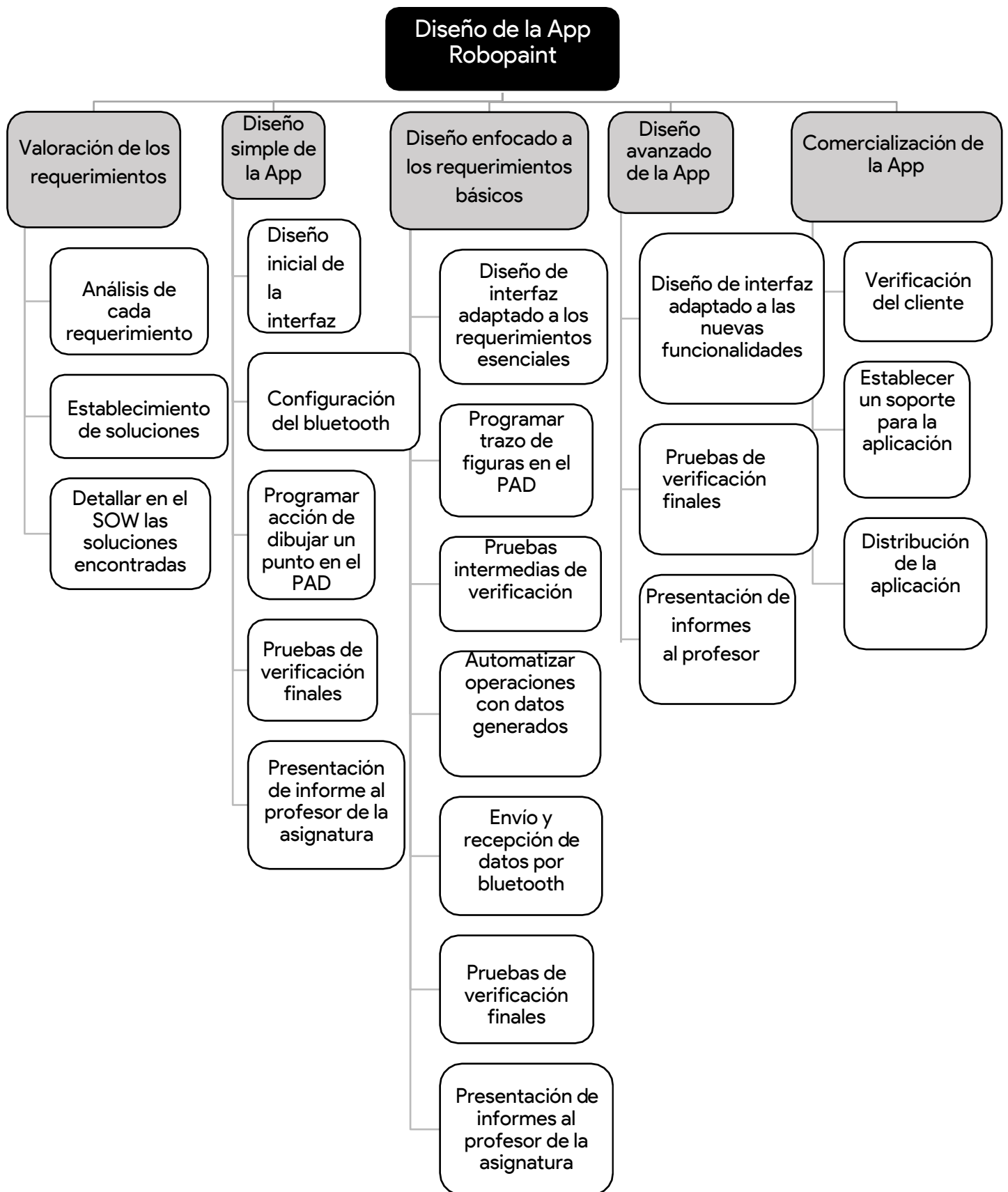
- 1. El presupuesto máximo del proyecto es de 80€.
- 2. El tamaño del brazo robótico impide trabajar en entornos muy grandes. Ejemplo: No será posible dibujar sobre formatos A1 y A0.

WBS General

Este proyecto contiene dos partes claramente diferenciables. A continuación, se expone el WBS del sistema de manera global; posteriormente se desarrolla el WBS de cada parte, explicando cada subsistema.



Subsistema de diseño de la aplicación: WBS y desarrollo



1. Valoración de los requerimientos

El objetivo de este conjunto de tareas consiste en establecer una serie de requerimientos por parte del equipo encargado del subsistema asociado a la aplicación que va a controlar el robot, así como las soluciones que se han adoptado para satisfacer dichos requerimientos.

Esta serie de tareas será ejecutada por todos los miembros del equipo encargado de diseñar la App coordinada con el otro equipo de ingenieros.

1.1. Análisis de cada requerimiento

En esta tarea se establecen los requerimientos que debe tener la aplicación para satisfacer las necesidades del cliente. En este caso, puesto que se trata de un caso puramente didáctico, los clientes serían los profesores de la asignatura. Donde se mantuvo una reunión con los clientes (profesores) para dar el visto bueno a los requerimientos adoptados, ya mencionados en el apartado requisitos del sistema.

1.2. Establecimiento de posibles soluciones para los requerimientos

Una vez se concretaron los requerimientos de la App, y fueron acordados con los profesores de la asignatura, es necesario establecer las soluciones para dichos requerimientos, y coordinar la organización y división de tareas por parte de los miembros del grupo encargado de realizar la App en otros subgrupos, con el objetivo de mejorar la eficiencia a la hora de ejecutar las tareas asociadas directamente al desarrollo de la App.

1.3. Detallar en el SOW todas las soluciones encontradas

Por último, a continuación de haber establecido las soluciones para llevar a cabo este subsistema. Se procederá a plasmar en el SOW toda la información, recogida por los miembros del grupo, necesaria para acometer el subsistema asociado a la App.

2. Diseño simple de la App

En esta tarea se pretende realizar un prototipo sencillo de la aplicación que cumpla los requerimientos más simples, para ello se va a recurrir al IDE Android Studio como entorno de programación.

2.1. Diseño inicial de la interfaz

Consiste en la creación de una interfaz inicial, que lejos de ser la interfaz final va a suministrar los elementos más básicos gráficos que debe tener la aplicación, como puede ser el PAD de dibujo o la gestión de la conexión por Bluetooth con el robot por parte del usuario. Para ello se va a hacer del metalenguaje XML, aunque se va a trabajar con método gráfico facilitado por la IDE utilizada.

Guillermo García Domínguez será el encargado de realizar esta tarea.

2.2. Configuración del bluetooth

Esta tarea cumple un requerimiento básico como es la conexión por bluetooth con el microcontrolador que gestiona finalmente el robot, por lo que va a ser necesario una comunicación fluida con el equipo encargado de la programación de dicho microcontrolador, para establecer los parámetros necesarios que determinan dicha conexión por bluetooth, y así garantizar la conexión del terminal con el subsistema del robot.

Esta tarea va a ser realizada por el miembro Antonio Cejas Laurel.

2.3. Programación de la acción de dibujar un punto en el PAD

Una vez queda diseñado gráficamente el PAD sobre el que se va a hacer el trazado de figuras que finalmente va a dibujar el robot, se va a pasar a programar la parte lógica que va a hacer posible la acción de poder dibujar un punto en la pantalla, donde para la programación de la actividad (pantalla donde se va a dibujar) se va a emplear el lenguaje de programación de Java. Una vez conseguido este objetivo, va a servir de inspiración, así como de motivación para cumplir los objetivos más avanzados, ya que va a dar a pie al trazo de figuras completas en el PAD, con lo cual es un objetivo de gran importancia.

Esta tarea va a ser realizada por los miembros Pedro José Díaz García y Raúl Olmo Ruiz.

2.4. Pruebas de verificación finales

Esta tarea va a consistir en comprobar el funcionamiento de lo que se ha programado con el uso de un Smartphone, así como un microcontrolador con un módulo bluetooth facilitados por la empresa.

Esta tarea va a ser llevada a cabo por todos los miembros del grupo encargado del diseño de la App.

2.5. Presentación de informe al profesor de la asignatura

Esta tarea va a consistir en realizar los documentos pertinentes donde se detallen todas las tareas realizadas anteriormente, además de los objetivos conseguidos hasta la fecha al profesor José Ángel Acosta Rodríguez.

Esta tarea va a ser llevada a cabo por todos los miembros del grupo encargado del diseño de la App.

3. Diseño enfocado a los requerimientos básicos

En este conjunto de tareas se va a proceder a un diseño más avanzado de la aplicación, que va a cumplir todos los requerimientos esenciales de la aplicación, como puede ser el trazo completo de figuras en el PAD, así como el envío de las referencias necesarias al subsistema del robot.

3.1. Diseño de interfaz adaptado a los requerimientos esenciales

En esta tarea se recurre a diseñar una interfaz más completa que aborde ciertas

animaciones, así como un diseño mucho más llamativo y atractivo para el cliente.

Esta tarea va a ser realizada por Guillermo García Domínguez y Antonio Cejas Laurel.

3.2. Programación de trazo de figuras en el PAD

Esta tarea supone una continuación de la anterior tarea de dibujar un punto, pero en este caso supondría el dibujo de múltiples puntos formando figuras continuas en el PAD, así como la posibilidad de poder realizar trazos discontinuos. Una vez sea posible la acción de dibujar un punto va a ser inmediato el poder dibujar un trazo, ya que consiste en múltiples puntos dispuestos de manera ordenada.

Esta tarea va a ser realizada por Pedro José Díaz García y Raúl Olmo Ruiz.

3.3. Pruebas intermedias de verificación

Consiste en realizar diversas pruebas para corroborar el correcto funcionamiento de la App, utilizando de nuevo un microcontrolador Arduino y un módulo bluetooth.

Esta tarea va a ser realizada por todos los miembros del grupo encargado del diseño de la App.

3.4. Automatizar las operaciones con los datos generados por las figuras dibujadas y por otros elementos de la aplicación

Esta tarea consiste en la programación de todas las operaciones lógicas, así como numéricas con el fin de obtener en cada momento las coordenadas tanto en el eje x como en el eje y, que permitan situar cada punto que compone la figura dibujada en el PAD por el usuario, así como su adaptación al plano donde tiene que dibujar el robot. Además de otros datos necesarios como el modo en el que se desea dibujar, es decir si en ese momento se quiere dibujar en modo continuo o en modo discontinuo, así como órdenes directas de encendido y apagado del robot, o bien de inicialización del robot.

Esta tarea va a ser realizada por Pedro José Díaz García y Raúl Olmo Ruiz.

3.5. Envío y recepción de datos por bluetooth al robot

Consiste en la programación de objetos y clases necesarias para el almacenamiento de datos en vectores y el envío de estos (lo cual hay que tener en cuenta que se pretende hacer en tiempo real), de modo que estos sean compatibles con el microcontrolador, es decir que puedan ser entendidos por el subsistema del robot. Además de esto también será necesario la recepción de datos por parte del subsistema del robot, y para ello va a ser necesario crear una serie de instrucciones que puedan almacenar dichos datos de forma adecuada para que la aplicación pueda operar con ellos, es decir pueda interpretar la información de dicho subsistema y actuar en consecuencia. Para esta tarea va a ser necesaria una comunicación constante y fluida entre los grupos de cada subsistema.

Esta tarea va a ser realizada por los miembros Guillermo García Domínguez y Antonio Cejas Laurel.

3.6. Pruebas de verificación finales

Consiste en la realización de diversas pruebas para corroborar que efectivamente se han cumplido los objetivos principales.

Esta tarea va a ser realizada por todos los miembros del grupo encargado del diseño de la App.

3.7. Presentación de informes al profesor de la asignatura

Esta tarea va a consistir en la realización de los documentos exigidos por el profesor José Ángel Acosta Rodríguez, en la fecha especificada, donde se incluirá todo el progreso en lo que respecta al desarrollo de la aplicación hasta dicho momento.

Esta tarea será realizada por todos los miembros del equipo encargado del desarrollo de la aplicación.

4. Diseño avanzado de la App

En este conjunto de tareas se pretende cumplir con una serie de requisitos considerados secundarios, puesto que la tarea principal que debe realizar el robot es dibujar correctamente el trazo realizado en la app, aunque aportan ciertas funcionalidades que otorgan un extra de calidad al proyecto elaborado.

4.1. Diseño de interfaz adaptado a las nuevas funcionalidades

El hecho de incorporar nuevas funcionalidades viene acompañado con una modificación, o una adaptación de la interfaz a esas nuevas funcionalidades, por lo que va a ser necesario añadir ciertos elementos que permitan la interacción del usuario con el uso de esas funcionalidades, de la forma más ergonómicamente posible.

Esta tarea va a ser realizada por todos los miembros del grupo encargado del diseño de la aplicación.

4.2. Pruebas de verificación finales

Consiste en la realización de determinadas pruebas para verificar que se cumplen correctamente todos los requisitos exigidos por el cliente. Una vez cumplidas estas pruebas se puede dar por concluido el diseño de la aplicación. Estas pruebas de verificación se harían en conjunto con el grupo encargado del diseño del subsistema del robot.

Esta tarea va a ser realizada por todos los miembros del grupo encargado del diseño de la App.

4.3. Presentación de informes al profesor de la asignatura

Esta tarea va a consistir en la redacción de una serie de informes pedidos por el profesor José Ángel Acosta Rodríguez con el fin de comunicarle todos los progresos realizados hasta la fecha en la que se realice el informe.

Esta tarea será realizada por todos los miembros del equipo encargado del desarrollo de la aplicación.

5. Comercialización de la aplicación

Una vez dadas por válidas las últimas pruebas de verificación habría que concretar una reunión con el cliente para que dé su aprobado a la aplicación.

Esta fase tendría lugar si se tratara de una empresa real, pero para este caso en el que se está llevando a cabo un proyecto meramente didáctico no llevarán a cabo esta serie de tareas.

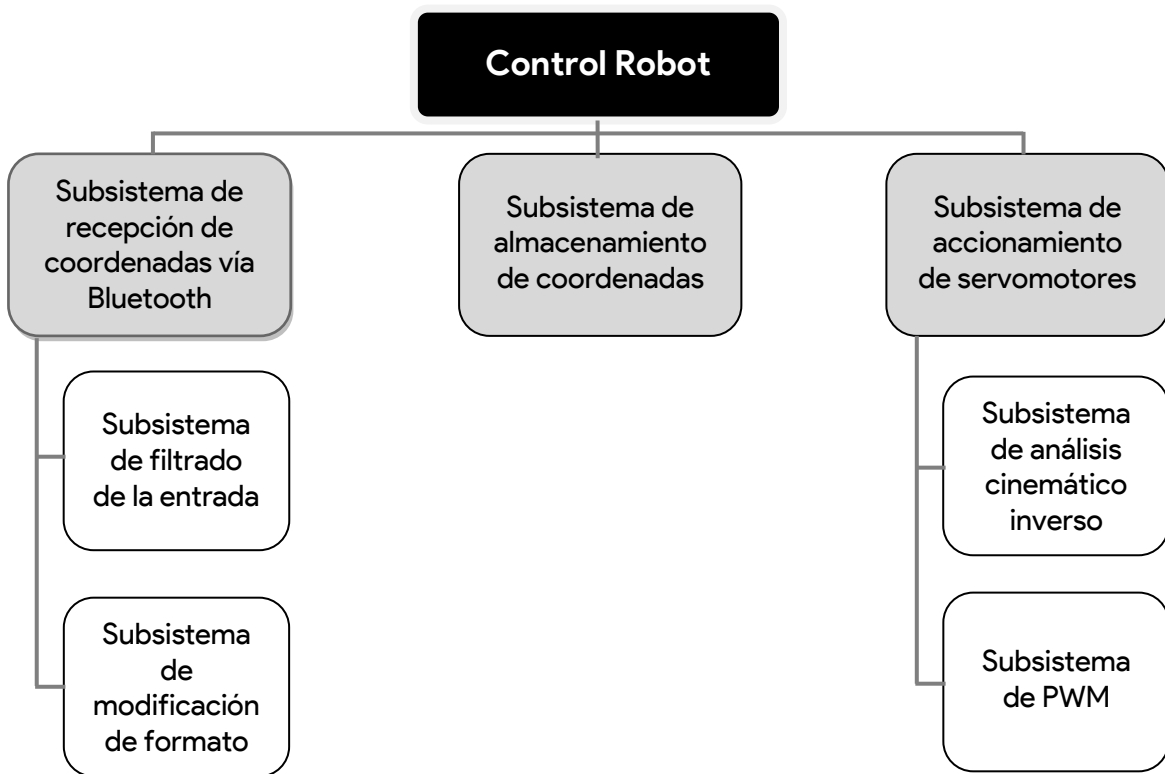
5.1. Establecimiento de soporte para los usuarios

Para la corrección de posibles errores o bugs de la aplicación, sería necesario un equipo de ingenieros que dé soporte a nuevas actualizaciones donde se corrijan dichos fallos, así como la realización de ciertos manuales para dar a conocer el funcionamiento de la aplicación.

5.2. Distribución de la App

En esta tarea habría que hacer llegar a los usuarios la aplicación con la que se va a controlar al robot, lo cual puede llevarse a cabo a través de un código QR, que nos enlaza a una carpeta de Google Drive, en la que se encuentra la apk disponible para su uso gratuito.

Subsistema de control: WBS y desarrollo



De mutuo acuerdo con el cliente, se establecen los puntos básicos que este proyecto debe cumplir para ser considerado satisfactorio. Dichas condiciones deben ser, a la vez, beneficiosas y suficientes para el cliente, y viables de realizar. Tras tener esto en cuenta, se ha llegado a la conclusión de que se ha de priorizar con creces la precisión del trazo del robot antes que la velocidad de éste.

Si bien el proyecto está ideado para reproducir el dibujo hecho por un ser humano en tiempo real, la limitación de la velocidad de los servomotores estará siempre presente. Como la velocidad del dibujo depende en gran parte del usuario, puede darse tanto el caso de que el robot sea capaz de seguir, al mismo tiempo, el trazo del usuario o que no sea capaz.

En este último caso es de vital importancia almacenar los datos que sean necesarios para poder finalizar la tarea, si bien la aceleración del proceso queda sujeto a un punto no mandatorio, sino opcional y a incluir como posible mejora si el tiempo lo permitiese.

El propio control del brazo ha sido dividido, a su vez, en, subsistemas más reducidos. Esto es así, aparte de por la propia organización del proyecto, debido a que es coherente con la realización de este. El propio control del robot, realizado vía software, estará dividido en funciones internamente. Esto es así ya que proporciona una gran ventaja, y es que realizar una depuración o modificación del control es mucho más fácil y seguro. Así, futuras versiones de iDraw serán más viables y factibles.

A continuación, se exponen los subsistemas que aparecen representados en el esquema de control del Robot, detallando su significado y su función:

1. Subsistema de recepción de coordenadas vía Bluetooth

Encargado de transformar un vector de datos de entrada desde el módulo Bluetooth, el HC06, en datos separados según su utilidad para el software y para el propio robot. Es el punto de unión entre los dos grandes subsistemas que conforman el proyecto: la app y el propio robot. Dentro del mismo podemos encontrar a su vez otros subsistemas bien diferenciados:

1.1. Subsistema de filtrado de entradas

El sistema de transmisión sin cable nos proporciona una serie de beneficios que, en el caso concreto de iDraw, son fundamentales. Sin embargo, también hay que tener en cuenta que existen ciertos inconvenientes a tener en cuenta. Uno de ellos es la posibilidad de recibir datos erróneos o pérdida de datos. Para solucionar esto, y teniendo en cuenta que se conoce el formato y rango de los datos de entrada, se puede descartar aquellos datos que no cumplan alguna de las características definidas. Si bien es cierto que este filtrado no es infalible, pero unido a un buen código de envío de datos, nos proporciona un porcentaje de fiabilidad más que suficiente.

1.2. Subsistema de modificación de formato

Una vez se ha obtenido el vector correcto, es importante dar formato y tratar los datos recibidos para que sean útiles. Es importante conocer que el vector de entrada no solamente transmite datos de posición, sino que también da información de ciertos parámetros de acciones auxiliares o internos. Por tanto, es de crucial importancia extraer los datos “auxiliares”, es decir, aquellos no referidos a posición, antes de empezar a tratar el formato del vector de posiciones.

Este subsistema será llevado a cabo por Álvaro Sánchez Cano.

2. Subsistema de almacenamiento de coordenadas

Una vez tengamos exclusivamente que tratar el vector de posiciones, es vital para el manejo del robot contar con un nuevo vector en el que almacenar x veces el vector de posiciones. Así, se busca conseguir que el robot pueda realizar una trayectoria con las coordenadas almacenadas en el nuevo vector.

Este subsistema será llevado a cabo por Álvaro Sánchez Cano.

3. Subsistema de accionamiento de servomotores

Tras conocer exactamente en qué posiciones se debe pintar y en cuales no, es necesario mover el brazo robótico hasta dichos puntos. Para ello, se deben realizar varias tareas, que han sido divididas en subsistemas a su vez:

3.1. Subsistema de análisis cinemático inverso

Las posiciones que rellenan nuestra pila están referidas a las posiciones del folio que deben ser pintadas. Es decir, los puntos que debe recorrer el extremo del brazo robótico. Sin embargo, la actuación del microcontrolador será llevada a cabo sobre los servomotores, hecho por el cual las posiciones de estos para llevar el extremo del

brazo robótico a la posición deseada serán, a priori, desconocidas. Esta es la tarea de este subsistema. A través de un análisis físico, se han de trasladar las posiciones ya obtenidas a seis posiciones, una para cada servomotor, de forma que la punta del rotulador pueda alcanzar todos los puntos deseados.

3.2. Subsistema de PWM

Una vez ya sabemos todas las posiciones a las que debemos llevar los servomotores a ellas. Para ello, se debe hacer la última traslación del proyecto, se debe traducir la posición a la que se debe llevar cada servomotor con la señal que debemos darles a los mismos para que lleguen a ellas. La técnica utilizada para esta tarea será el PWM (Pulse Width Modulation), que permitirá controlar cada servomotor con tan solo un pin del microcontrolador. Esto permite reducir considerablemente el precio de este, y además ahorrar fallos, debido a la robustez de este método y al ahorro de cableado que se realiza.

Este subsistema será llevado a cabo por Javier Gómez Jiménez.

Periodo y planificación

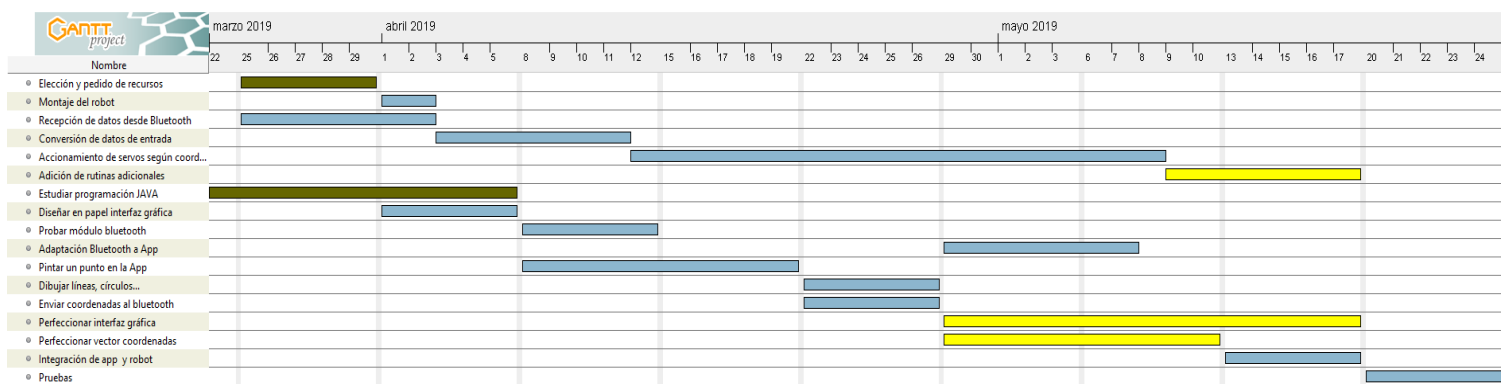
Periodo de trabajo

El periodo establecido va a venir dado por el tiempo que nos otorgue los profesores de la asignatura de Proyectos Integrados, que en este caso actuarían como clientes que nos han encargado la realización de este proyecto.

Por tanto, el periodo de trabajo comienza el 22 de marzo, finalizando el 27 de mayo. En el diagrama de Gantt se puede ver más explícito todo el periodo.

Diagrama de Gantt

A continuación, se expone el diagrama de todo el proyecto:



Hay tareas incluidas dentro de otras.

Las tareas que aparecen en verde son las tareas previas, o no específicas del proyecto. Las azules son las básicas. Por último, las amarillas son opcionales de perfeccionamiento del sistema.

Análisis de riesgos

El proyecto, según la parte que se quiera analizar, puede tener una serie de riesgos que impidan el desarrollo correcto de este o su funcionamiento. Para ello, vamos a dividir este apartado en tres grupos, denominados riesgos estructurales, de programación en JAVA o de control.

1. Riesgos estructurales

Dentro de los riesgos estructurales podemos tener diferentes tipos:

1.1. Piezas defectuosas

Una vez comprado nuestro brazo robótico, nos podemos encontrar con que algunas piezas incluidas vengan inservibles o se hayan podido romper durante el envío. La probabilidad de que esto ocurra puede variar según el tipo de envío y como venga revestido el paquete, por lo que podemos estimar que puede ser poco probable, ya que cada vez se mejoran los empaquetamientos de envíos y las formas de llevar a cabo este. La solución para este caso, lo primero, es que debemos realizar la compra del brazo lo antes posible para evaluar su estado y así poder decidir sobre qué hacer a partir de lo visto. Una vez que el objeto traiga piezas defectuosas o algo roto, deberíamos pedir la devolución de este con la máxima antelación posible; y si no se pudiera, se pediría una reclamación para la devolución del dinero.

1.2. Entrega del brazo

Otro caso posible es que nuestro objeto no nos llegue para poder realizar el proyecto. La probabilidad para este tipo de problema es muy poco probable, ya que podemos asegurar que la empresa distribuidora Amazon trabaja muy bien en sus envíos y es muy poco probable que alguno se extravíe. Entonces, como bien está descrito en el apartado anterior, si esto ocurriera, necesitamos pedirlo a la casa distribuidora con bastante antelación para poder evitar cualquier situación de esta y poder reclamar que el producto no se ha entregado.

1.3. Rotura de componentes

Durante el montaje y prueba del brazo robótico puede ocurrir que alguno de sus componentes puede quedar inutilizable debido a una mala configuración o alguna situación incontrolable. La probabilidad para este problema puede ser mayor que en los casos anteriores, clasificándolo con una probabilidad media, ya que tenemos que probar nuestro proyecto en muchas ocasiones y en una de estas puede ocurrir dicho problema. Para poder resolver esto, hemos dejado una parte del presupuesto destinado a este posible problema, y por si ocurre, debemos de tener los elementos que los reemplazan lo antes posible, por lo que debemos pedirlo con antelación.

1.4. Rotura del brazo robótico

Es posible que, a lo largo de la realización del proyecto, nuestro brazo comprado quede inservible por algunas situaciones como caída, quemadura, etc. entre otros. La probabilidad es baja para que suceda, ya que si el conjunto humano que forman el grupo de trabajo es responsable y sabe cómo funciona el objeto y lo que no se le puede hacer, es muy poco probable que ocurra esto. Si esto sucede, se recurriría a la realización de un modelo de brazo en el software CATIA V20 para su posterior impresión 3D y montaje.

2. Riesgos en la aplicación

Dentro de este apartado referido a la programación de la aplicación en lenguaje JAVA, podemos encontrar los siguientes problemas:

2.1. Error interno en la aplicación

En algunas situaciones donde estemos probando la app con el robot, esta puede tener un error interno que haga que todo se quede congelado y por tanto no se pueda hacer nada. Este tipo de riesgos tiene una probabilidad alta de suceder, ya que el código de programación puede fallar en cualquier momento y por cualquier cosa. Para poder solucionar este tipo de problemas, lo primero es hacer ir al robot al estado inicial de reposo, y desde ahí hacerlo funcionar otra vez porque puede haber sido un error puntual; también debemos de poner la app en varios dispositivos ya que si uno deja de funcionar tenemos que tener otro para poder seguir con el desarrollo del proyecto. Para mitigar aún más este riesgo, se va a utilizar un depurador para corregir todos los errores internos. Lo más importante de esto es verificar cada código que se implemente para así tener los mínimos errores.

2.2. Pérdida de código

Puede ocurrir que mientras se implemente el código haya alguna pérdida debido a que el desarrollador o el sistema con el que se esté desarrollando pueda tener un error que haga que todo lo que no se haya guardado quede eliminado. Esto tiene una probabilidad baja de que ocurra ya que el programador sabe y debe asegurar que el programa debe de estar guardado y sincronizado en cada instante. Para evitar esto, se debe hacer una copia de seguridad a cada instante de modificación y tenerlo guardado en diferentes lugares para que no se pierda en ningún momento. Para ello se utilizará la herramienta GIT. Con esta herramienta desaparece por casi por completo este riesgo.

2.3. Interferencia Bluetooth

Mientras estemos probando nuestro proyecto, el cual está conectado mediante bluetooth a nuestro dispositivo, puede ocurrir que haya una interferencia entre el brazo robótico y nuestro dispositivo de control. La probabilidad de esto es de una probabilidad media, ya que puede ocurrir que algún dispositivo se entrometa en la conexión que tengamos produciendo dichas interferencias, por ejemplo. Para que esto no ocurra, debemos comprobar en todo momento el estado del enable de nuestro robot, y a través de este sabemos si hay alguna interferencia o no.

3. RIESGOS EN EL CONTROL

En este apartado referido al control del robot, podemos encontrar algunos riesgos como los siguientes:

3.1. Falta de memoria en el controlador

Puede ocurrir que, una vez diseñado el código, nuestro microcontrolador (Arduino) no disponga de suficiente memoria RAM para ejecutar dicho código tenga una capacidad menor en su memoria que impida que el código se pueda ejecutar correctamente. Existe una probabilidad media de que ocurra esto, ya que nuestro programador debe saber y tener siempre en cuenta que nuestro código debe encontrarse optimizado al máximo, reduciendo así la posibilidad de saturar al

microcontrolador.

3.2. Error en el controlador

Nuestro controlador puede sufrir diversos daños según lo que le implantemos que queremos que haga. Entonces, puede ocurrir que nuestra placa de Arduino se queme o se rompa si nos equivocamos metiendo algunos parámetros de diversa importancia. Tiene una probabilidad media, ya que nuestro grupo con el tema de los elementos de desarrollo siempre es muy cauto y nos dedicamos a probar lo máximo posible hasta alcanzar un valor óptimo de funcionamiento sin inutilizar nada. Como solución al problema si este sucede, debemos tener placas de repuestos listas para sustituir la defectuosa.

3.3. Descontrol del robot

Este tipo de riesgos aparecen en las pruebas intermedias del robot con una probabilidad media. Esto sucede porque se está probando un código incompleto, con huecos que son responsables de comportamientos imprevistos del robot. Aún así, este tipo de riesgos se van reduciendo a la par que avanza el proyecto, quedándose muy reducido al final del proyecto.

Calendario de entregables

- **Entregable 1:** Informe sesión 1.0. Fecha: 28/04/2019
- **Entregable 2:** Informe sesión 2.0. Fecha: 12/05/2019
- **Entregable 3:** Informe sesión 3.0. Fecha: 19/05/2019
- **Entregable 4:** Presentación y SOW final. Fecha: 27/05/2019

Criterios de aceptación

Criterios de aceptabilidad

El proyecto se dará por finalizado cuando se cumplan los siguientes requisitos:

- Correcto funcionamiento del brazo.
- Dibujo de trazos continuos y discontinuos.
- Funcionamiento en tiempo real.
- Sensibilidad a la velocidad de escritura.
- Tiempo de retardo.
- Recorrido del espacio de trabajo.
- Rango de operación deseado.
- Diseño aceptable del brazo robótico.
- Diseño aceptable de la App.
- Modo variable de funcionamiento.
- Sujeción suficiente de la garra.
- Control correcto por bluetooth.
- Aplicación en varios sistemas operativos.
- Aplicación disponible a través de un código QR.
- Tiempo de preparación y calibración del brazo mínimos.
- Tiempo inicio y reinicio de la aplicación mínimos.
- Precisión de la aplicación.
- Velocidad y precisión del brazo robótico.
- Velocidad de transmisión por bluetooth aceptable.

Además, existen otras normas que se podrían añadir de carácter secundario u opcional; dado que presentan este carácter no obligatorio, no se han añadido a los criterios de aceptabilidad.

Matriz de verificación

| Requisito | Nombre Requisito | Verificación | | | | Nombre prueba | Estado |
|------------|---|--------------|---|---|---|---------------|------------|
| | | I | A | D | T | | |
| F.1 | Funcionamiento del brazo | | | X | X | Test 1 | Completado |
| F.2 | Dibujo de trazos continuos y discontinuos | | | X | X | Test 2 | Completado |
| F.3 | Funcionamiento en tiempo real | | | X | X | Test 2 | Completado |
| F.4 | Sensibilidad a la velocidad de escritura | | | X | X | Test 2 | Completado |
| P.1 | Tiempo de retardo | | | X | X | Test 1 | Completado |

| Requisito | Nombre Requisito | Verificación | | | | Nombre prueba | Estado |
|-----------|--|--------------|---|---|---|--|------------|
| | | I | A | D | T | | |
| P.2 | Recorrido del espacio de trabajo | | | X | X | Test 1 | Completado |
| D.1 | Rango de operación | | | X | | Alcance del brazo robótico | Completado |
| D.2 | Diseño brazo robótico | X | | | | Desmontaje de los componentes del brazo robótico | Completado |
| D.3 | Diseño aplicación | X | | X | X | Test 2 | Completado |
| D.4 | Modos funcionamiento aplicación | | | X | X | Test 2 | Completado |
| D.5 | Sujeción garra | | | X | X | Test 1 | Completado |
| O.1 | Control por bluetooth | | | X | X | Test 2 | Completado |
| O.2 | Aplicación en varios dispositivos | X | | | | Comprobación de la aplicación instalada en varios dispositivos | Completado |
| O.3 | Aplicación disponible en Google Drive a través de un código QR | X | | | | Instalación de la apk a través del código QR | Completado |
| O.4 | Tiempo preparación y calibración del brazo | | | X | X | Test 1 | Completado |
| O.5 | Tiempo inicio y reinicio aplicación | | X | X | | Inicio y reinicio de la aplicación | Completado |
| S.1 | Precisión aplicación | | | X | X | Test 2 | Completado |
| S.2 | Velocidad y precisión del brazo robótico | | | X | X | Test 2 | Pendiente |
| S.3 | Velocidad transmisión bluetooth | | | X | X | Test 2 | Completado |

Plan de pruebas

Las pruebas para realizar a nuestro sistema son:

| Número del test | 1 |
|-----------------------------------|---|
| Tipo de test | Software y mecánica |
| Instalaciones donde se prueba | Nuestro centro de trabajo |
| Ítem probado | Brazo robótico |
| Procedimiento y duración del test | Para probar el brazo robótico, haremos siga unas coordenadas espaciales en su entorno de trabajo. La duración del test será de unos 20 minutos. |
| Duración de la campaña de test | 5 horas |
| Fecha de la campaña de test | Abril - Mayo |

| Número del test | 2 |
|-----------------------------------|--|
| Tipo de test | Software y mecánica |
| Instalaciones donde se prueba | Nuestro centro de trabajo |
| Ítem probado | Brazo robótico + aplicación |
| Procedimiento y duración del test | A través de la aplicación, se dibuja a través del pad de la app. Posteriormente, se transmiten dichas coordenadas al robot para realizar dichos trazos en su espacio de trabajo. La duración del test será de unos 20 minutos. |
| Duración de la campaña de test | 6 horas |
| Fecha de la campaña de test | Mayo |

Material y presupuesto

Material necesario

Para llevar a cabo el sistema se necesitan los siguientes materiales:

- Un brazo robótico programable con sus seis servos.
- Placas de Arduino UNO.
- Un módulo bluetooth HC-06.
- Material electrónico (protoboards, cableado, etc.).
- Computadoras para programar.
- Programas de software libre como *Arduino IDE 1.8.9* y *Android Studio 3.3.2*.

Presupuesto

Para la elaboración del presupuesto se han llevado a cabo dos criterios:

- Material necesario que el cliente obtiene con la compra del sistema, que actualmente no se posee.
- Material necesario que el cliente obtiene con la compra del sistema, donado al proyecto por terceros.

| Material necesario | Cantidad | Precio |
|---|----------|----------------|
| Brazo robótico programable 'Diymore ROT3U 6DOF' con 6 servos 'MG996R' | 1 | 61,99 € |
| Servo 'MG996R' de repuesto | 2 | 12,99 € |
| Soporte para el brazo robótico | 1 | 3€ |
| Total a pagar | | 77,98 € |

| Materiales donados/ya adquiridos | Cantidad | Valor |
|---|----------|-----------------|
| Placa 'Arduino Uno' | 3 | 12,90€ |
| Módulo Bluetooth HC-06 | 1 | 8,90 € |
| Material electrónico (protoboard, potenciómetro, cables...) | 1 | 10 € |
| Total donado/ya adquirido | | 33,80 € |
| Total acumulado | | 111,78 € |

Referencias

Figura 1: Esquema iDRAW

Fuentes de las 4 imágenes, de derecha a izquierda:

<https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR7EsNEw8gyPTVPeVX33QGrojS-46dNxG70hPWwbU1mpqM3ACrnpA>

https://i.blogs.es/c82c1f/arduino-bluetooth1/450_1000.png

<https://1.bp.blogspot.com/-9ScNli1TjV0/V5H4AqlqRRI/AAAAAAAAABlg/ywutt-FKwSAJaGUEKp5oNVamh29vN3TACLcB/s1600/bluetooth00.jpg>

https://cosmosdigitalart.com/wp-content/uploads/2018/08/tablet_Galaxy_Tab_S3_dibujo.jpg

Figura 2: e-David

Fuente de la imagen:

<http://robotsinsider.com/edavid-an-artist-robot-that-paints-on-paper/>

Conclusión

Como se especificó en la introducción, iDRAW está diseñado para un uso didáctico y educativo. El objetivo es permitir un primer acercamiento de los jóvenes a la robótica permitiendo controlar el robot desde una app de uso libre y gratuito para realizar diversas actividades. Nuestro proyecto se ha buscado centrar el diseño para pintura, soldadura o incluso un futuro prototipo para robots de operaciones quirúrgicas (asistente del robot Da Vinci, por ejemplo); con un presupuesto ajustado de 80€. Si se desea implementar este proyecto en un uso industrial que necesite una mayor precisión será necesario adquirir un brazo robótico con servos de mayor calidad, fuerza y precisión.