1. **DISEÑO DEL MANIPULADOR**

Para el diseño del manipulador se propuso un brazo robótico con un grado de libertad, orientado en forma perpendicular al robot y paralela a su eje diferencial, por lo cual, se logro optimizar posibles problemas de diseño reconocidos en la adición de juntas. Obteniendo el diseño tipo riel con el piñón cremallera, convirtiendo una fuerza rotacional en otra unidireccional, en conjunto de un endeffector con una orientación plana, mostrada en la siguiente figura:

Diagram

Description automatically generated

El endeffector planteado solo tiene una estructura de cierre otorgada por el servo dado dos estados fijos, como cerrado o abierto.

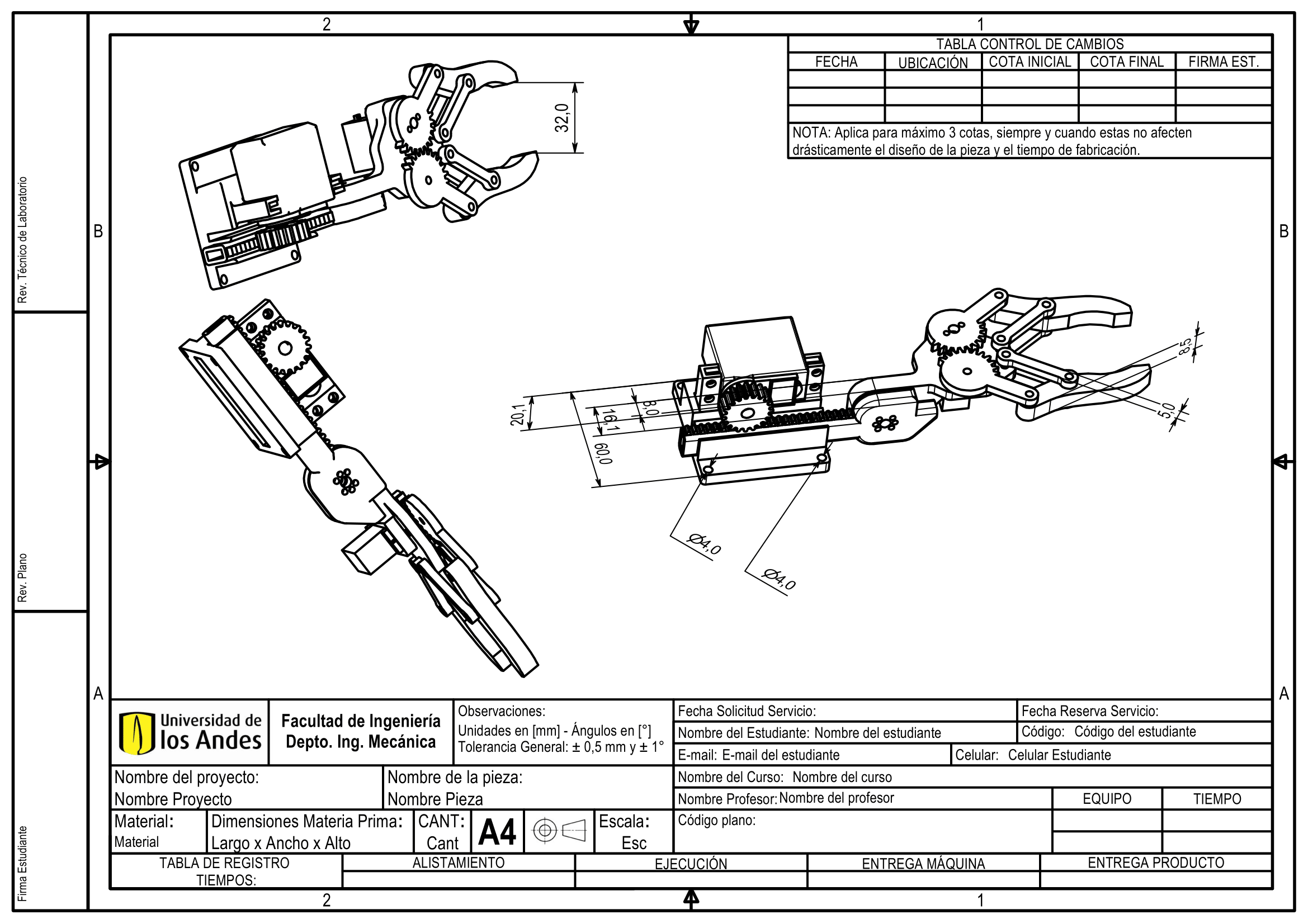
1. **LISTA DE MATERIALES**

Los materiales usados para su construcción fueron enteramente materia prima de impresión 3D, PLC en concreto, ofreciéndonos fáciles cambios de sección y maniobrabilidad a la hora de usar tonillos y uniones.

Para la parte electrónica se hizo uso de dos servos, uno de referencia sg90 y el otro mg995, diferenciados en mayor medida en su caja de reducción y tamaño de motor incluido.

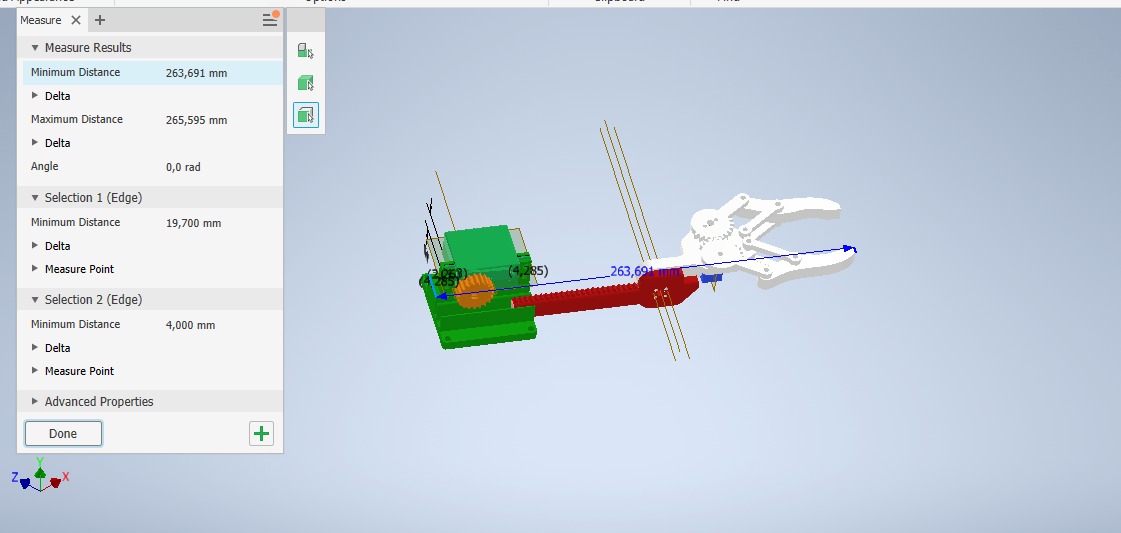
1. **PLANO MECÁNICO**

En los planos mecanicos se incluye las cotas principales del modelo, aparte de las vistas que permiten la replicacion de los planos, ya sea, por manufactura o por extrusion STL.

****

1. **VOLUMEN DE TRABAJO**

El espacio de trabajo máximo establecido es de 26.37cm, sin embargo, gracias a la ubicación y anclaje, el espacio de trabajo funcional de la misma corresponde de 9 a 15cm, relacionados con su máxima y mínima extensión, con un factor de seguridad loable.

****

1. **PLANO ELECTRÓNICO**

**Diagrama, Esquemático

Descripción generada automáticamente**

1. **INTEGRACIÓN PARTE MECÁNICA Y ELECTRÓNICA**

Para la integración de la parte mecánica, eléctrica se hizo uso de uniones en los ejes de rotación de los servos controlados por PWM, ajustando la velocidad a los desplazamientos posibles, en conjunto del diseño mecánico coherente.

1. **SOLUCIÓN TALLER**
2. **PUNTO 1**

En este punto se solicitaba permitir al usuario controlar por teclado el manipulador del robot por medio de un nodo en ROS llamado /robot\_manipulador\_teleop. Para esto se creo el topico /manipulador para que guarde la tecla que oprime el usuario y dependiendo esta se realice el movimiento de la siguiente manera: si se oprime la tecla “i” el manipulador se mueve hacia adelante, si se oprime la letra “k” el manipulador se mueve hacia atrás, si se oprime la tecla “z” el manipulador se cierra y si se oprime la letra “x” el manipulador se abre.

A continuación, se muestra un esquemático de ROS que resume la solución implementada.

Diagrama

Descripción generada automáticamente

1. **PUNTO 2**

El segundo inciso requería que se graficara la posición del end-effector en tiempo real. Para explicar su funcionamiento, es necesario recalcar que el end-effector estaba compuesto de dos servos y que el único grado de libertad que se empleó para este taller era manipulado por un solo servo. Dichos servo-motores eran controlados a su vez por el Arduino, a partir del cual (mediante el uso de ROS-Serial) se publicaba el ángulo que poseía el servo en tiempo real por un tópico de tipo Int llamado *“/angle”*.

A partir del ángulo, se empleó la interfaz gráfica construida para el Taller 2, en donde se agregó un suscriber a dicho tópico y se graficó la posición del end-effector en el marco de referencia local a partir de hallar la proporción que representaba un desplazamiento angular en términos de desplazamiento lineal del end-effector.

Los resultados obtenidos se muestran a continuación:

* **Posición inicial del end-effector**

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

* **Posición final del end-effector**

Gráfico, Gráfico de rectángulos

Descripción generada automáticamente

Por otro lado, la topología de ROS que describe este inciso se presenta a continuación:

Diagrama

Descripción generada automáticamente

1. **PUNTO 3**

En este punto se solicitaba que dada una posición ingresada por un usuario el end-effector se desplace de manera autónoma a esta posición. Para esto se creo el nodo /robot\_manipulador\_planner que recibe la posición (en cm) deseada por el usuario por medio del tópico /servo, y calcula el ángulo que debe desplazarse el servo a partir de la siguiente ecuación:

Esta ecuación se obtiene ya que se la posición de referencia es el centro del end-effector, por lo que cuando el brazo esta comprimido el end-effector se encuentra a 10 cm del robot, y, además, la máxima distancia que se puede contraer el manipulador es de 5cm, por lo que la posición del end-effector debe estar entre 10cm y 15cm.

A continuación, se muestra un esquemático de ROS que resume la solución implementada.

Diagrama

Descripción generada automáticamente

1. **PUNTO 4**

En este punto se solicitaba que dada una estructura de madera donde se encontraban 3 ping pong de diversos colores, el robot fuera capaz de agarrar el del color especificado a través del tópico */robot\_manipulator\_ping\_pong.* Para esto, se desarrolló una máquina de estados en ROS a través del paquete de Smach. A continuación, se muestra un esquemático que representa la máquina de estados desarrollada.

Diagrama

Descripción generada automáticamente

En el estado “Waiting For Color” se espera a que el usuario publique en el tópico */robot\_manipulator\_ping\_pong* el color del ping pong que quiere agarrar. Una vez se publica el color se revisa que el color sea válido, si el color no es válido entonces se queda esperando otro color, en cambio si el color es válido se le envía la información del color a ser detectado al nodo de visión y se pasa al estado “Checking Color”*.*

En el estado “Checking Color” se publica una velocidad lineal a través del tópico */cmd\_vel* para que el robot se mueva en búsqueda del ping pong adecuado. Al mismo tiempo el nodo de visión está verificando 3 cosas: Que exista un objeto redondo, que el objeto redondo sea del color especificado y que dicho objeto se encuentre en justo el centro del campo de visión. Si se cumplen estas 3 condiciones, entonces, esté nodo publicará un “1” en el tópico, de lo contario pública “0”. En el momento que se publica “1”, entonces, el robot se detiene y se pasa al estado “Catch”.

En el estado “Catch” se pública “x” en el tópico *\manipulador* para que el end-effector de la garra se abra, posteriormente se pública al mismo tópico “i” durante 2 segundos (tiempo necesario para que el manipulador alcance su máximo estiramiento), después de esto se pública “z” para cerrar el end-effector y, por último, se pública durante 2 segundos “k” para volver a comprimir el manipulador. Una vez se termina esta secuencia de acciones, entonces, se da por terminada la ejecución de la máquina de estados.

A continuación, se muestra un esquemático de ROS que resume la solución implementada.

Diagrama

Descripción generada automáticamente