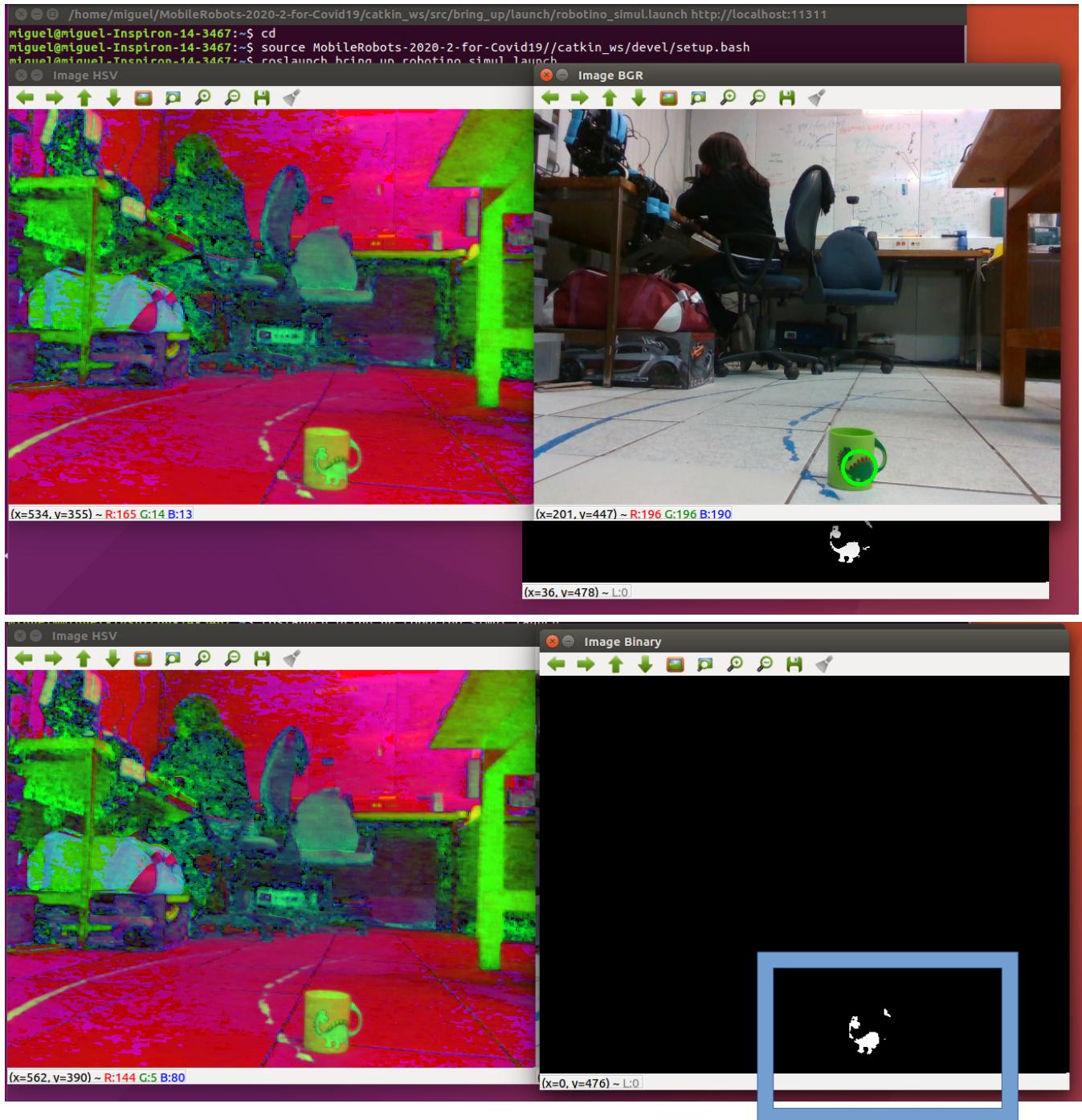


Práctica 6: Segmentación por color.

-Capturas de pantalla al ejecutar la simulación sin modificar ningún dato.



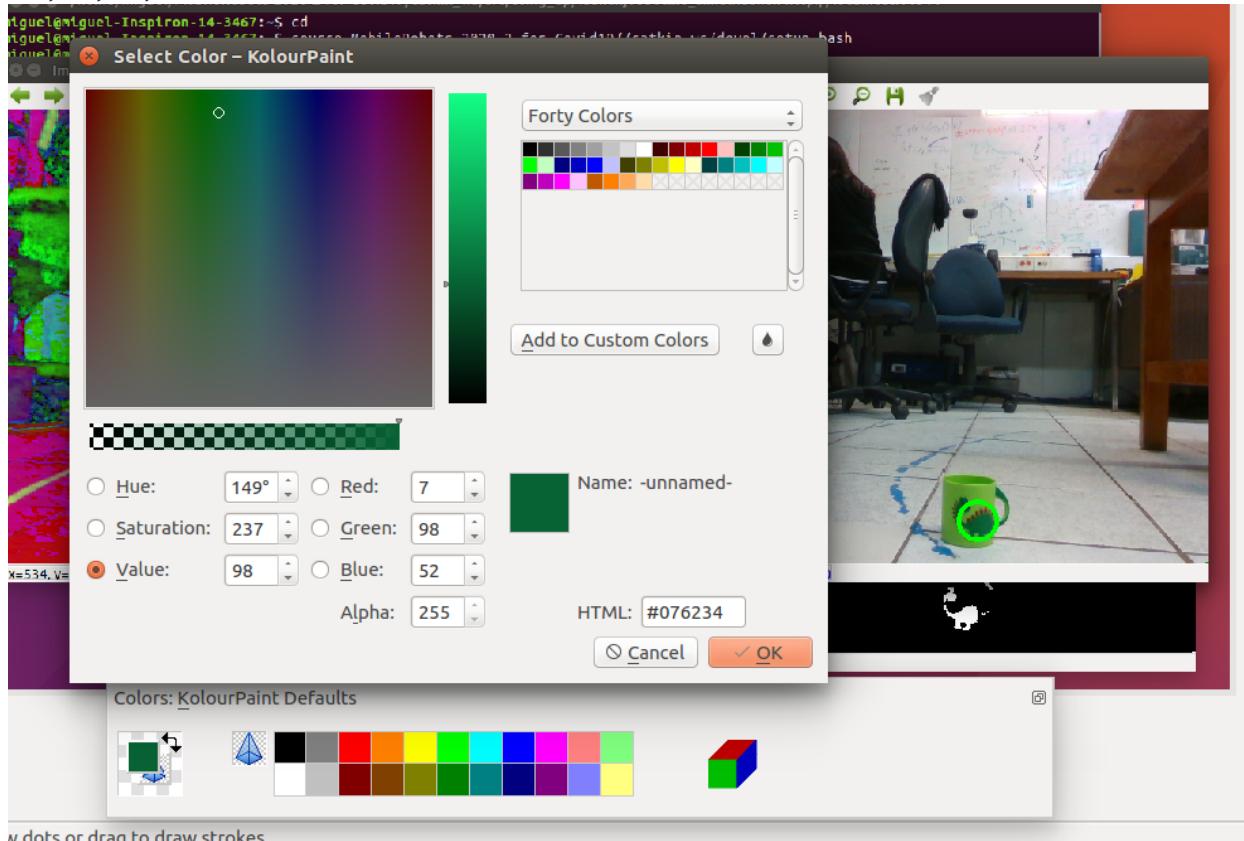
Se alcanza a mostrar la silueta del dinosaurio, pero la silueta de la taza no.

-Capturas de pantalla donde se muestra el valor HSV del color verde de la taza en KolourPaint, se tomaron varias muestras que modificaban el valor HSV pero siempre dentro de un rango.

H= 149°, 151°, 141°, 151°

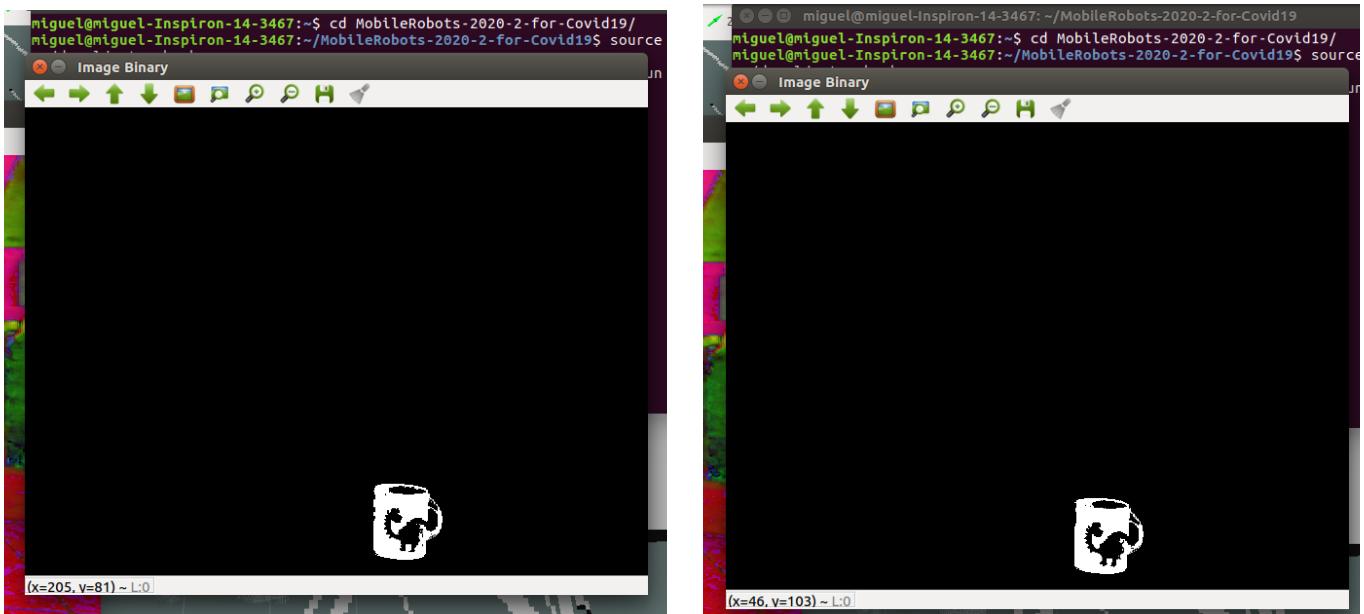
S= 237, 237, 213, 232

V= 98, 97, 84, 99



-Capturas de pantalla donde se muestra la segmentación de la lata de coca-cola o de la taza verde. Esto es modificando el archivo “color\_segmentation.py” y revisando la documentación de “([https://docs.opencv.org/3.4/da/d97/tutorial\\_threshold\\_inRange.html](https://docs.opencv.org/3.4/da/d97/tutorial_threshold_inRange.html))” para así determinar el umbral HSV en el cual se encuentra la taza. Los valores obtenidos se cambiaron y se ejecuto nuevamente la simulación obteniendo los siguientes resultados.

```
ef callback_rgb(msg):
    bridge = cv_bridge.CvBridge()
    img_bgr = bridge.imgmsg_to_cv2(msg, desired_encoding="bgr8")
    img_hsv = cv2.cvtColor(img_bgr, cv2.COLOR_BGR2HSV)
    img_bin = cv2.inRange(img_hsv, numpy.array([42,72,100]), numpy.array([83, 215, 255]))
    idx = cv2.findNonZero(img_bin)
    [centroid_x, centroid_y, a, b] = cv2.mean(idx)
    cv2.circle(img_bgr, (int(centroid_x), int(centroid_y)), 20, [0, 255, 0], thickness=3)
    cv2.imshow("Image BGR", img_bgr)
    cv2.imshow("Image HSV", img_hsv)
    cv2.imshow("Image Binary", img_bin)
    cv2.waitKey(1)
```



Se puede apreciar una mejor silueta del dinosaurio y de la taza, pero para llegar a dicha segmentación, se complicó encontrar los valores del umbral HSV, por lo cual llego un momento de modificar valores de uno en uno para llegar a una mejor apreciación, situación que aveces ayudaba pero en ocasiones empeoraba el resultado, teniendo como mejor resultado el que se muestra en las imágenes.

