1. **PLANIFICACIÓN:** Esto ya tal…
2. **CONTROL** (a falta de poner pasos anteriores de planificación): Sabemos la posición actual y a la que queremos ir -> Controlador a partir de x(k|k) y X\_ref nos dice las consignas de v y ω. (ALBERTIÑO)
3. Reseteamos la odometría y movemos al robot 0.01 segundos (o lo que corresponda) según las consignas de v y ω. (Esto se hace dentro del bucle infinito directamente o incluso dentro de la función anterior). Recordad darle un apoloUpdate para que se vean los cambios.
4. **PREDICCIÓN (EFK):** Con los valores de la v y la ω se calcula la matriz Q(k) (con Q\_pu, que se ha obtenido de la calibración) -> Función **Matriz\_Q** ya programada
5. A partir de las medidas de la odometría, que se pueden hacer fuera o dentro de la siguiente función, se calcula la posición nueva del robot X(k+1|k), así como la nueva matriz P(k+1|k) -> función **GetPositionFromOdometry** ya programada
6. Calcular la Z estimada. Para ello, se empieza por darle al robot (me refiero al robot de la clase creada de Matlab tan chula) la posición nueva estimada con la función robot = **actualizar\_posicion**(robot, X(k+1|k)). A continuación, se usa la función [z, H] = **estimar\_medidas**(robot, entorno) para obtener la z estimada y el jacobiano H. Esto ya está programado, pero hay que meterlo al bucle que WILLY tiene que hacer (filtro de Kalman).
7. **COMPARACIÓN (EFK):** Hacer todas las cosas de Kalman, que son 3 líneas de código, es decir, matriz S (usando la matriz R de la calibración que vamos a tener que inventarnos), matriz W y vector de innovación en la medida ν.
8. Distancia de Mahalanobis. Si es muy alta, ignoramos el resultado y nos quedamos con la odometría como estimación buena, o lo que haga falta. Pero, si sale muchas veces seguidas muy alta, va a haber que hacer algún tipo de control reactivo. De momento que se pare el robot y no haga nada. Otra opción es que otro tipo de controlador actúe en el siguiente ciclo, moviéndose hacia atrás un poco y replanificando y a ver si funciona o algo.
9. **CORRECCIÓN (EFK)**: Calcular la nueva X(k+1|k+1) y P(k+1|k+1).
10. Repetir. Sin replanificar, salvo que algún caso del paso 8 haga que sea buena idea.