

Proyecto Pepper

Andres Badillo, Ana Vargas, Cristian Olarte

25 de abril de 2025

1. Pepper coreograhp

1.1. Librerías

- **Librería qi:** Es el framework principal para programar robots Softbank/Aldebaran como Pepper. Permite la comunicación con los servicios del robot, gestión de memoria y módulos.
- **Librería argparse:** Facilita el procesamiento de argumentos de línea de comandos en Python. Útil para crear aplicaciones para Pepper que requieran parámetros configurables.
- **Librería os:** Permite interactuar con el sistema operativo. En Pepper, es útil para gestionar archivos, rutas y variables de entorno en el robot.
- **Librería almath:** Biblioteca matemática específica de Aldebaran/Softbank que facilita cálculos relacionados con la cinemática, transformaciones y posicionamiento del robot Pepper.
- **Librería math:** Proporciona funciones matemáticas estándar en Python, necesarias para cálculos en algoritmos de movimiento y posicionamiento.
- **Librería motion:** Controla los movimientos de Pepper, permitiendo programar desplazamientos, gestos y posturas.
- **Librería httplib:** Permite realizar peticiones HTTP, útil para que Pepper se comunique con servicios web externos.
- **Librería json:** Facilita el trabajo con datos en formato JSON, comúnmente usado para intercambiar información entre Pepper y servicios externos o para configuraciones.

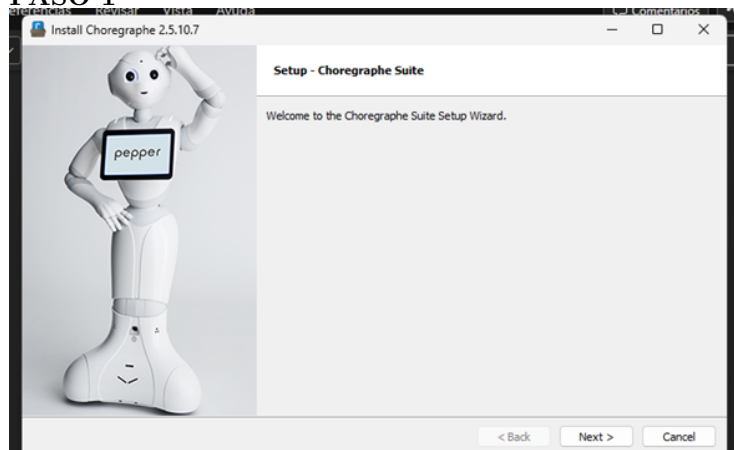
Para instalar "Choregraphe" debemos ingresar al siguiente link:

<https://mega.nz/file/N6IHZSbY#0om2GPPUeKcS-sY29pgRW5MZ4C01WAC2WEhZFh6V88>

Una vez instalado el programa debemos seguir los siguientes pasos:

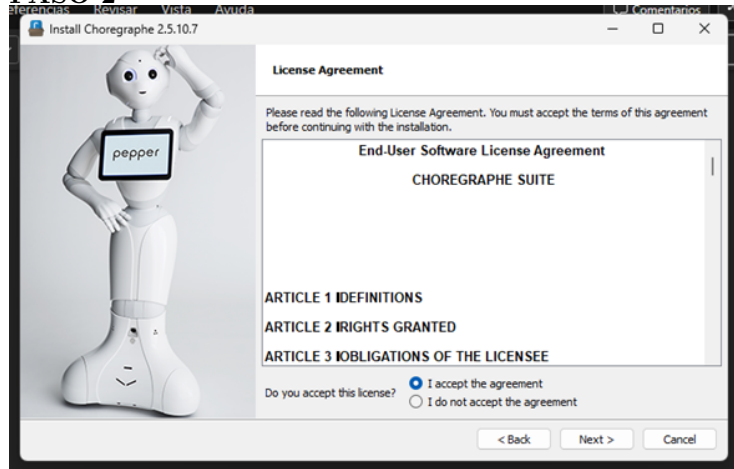
1.2. Pasos para la instalación de Choregraphe

PASO 1



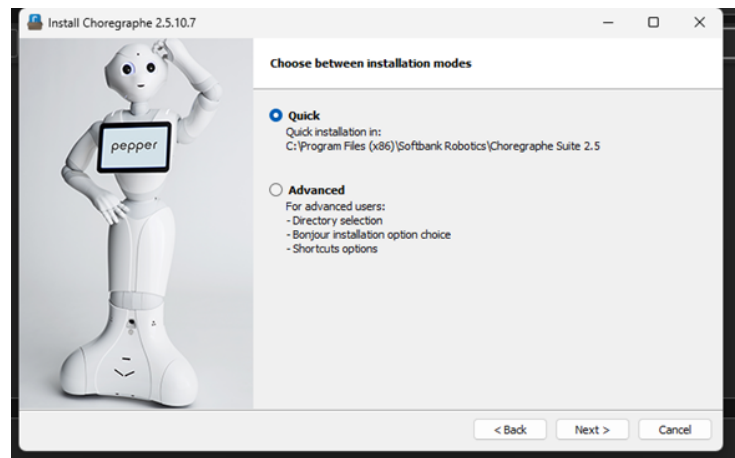
Le damos "Next".

PASO 2



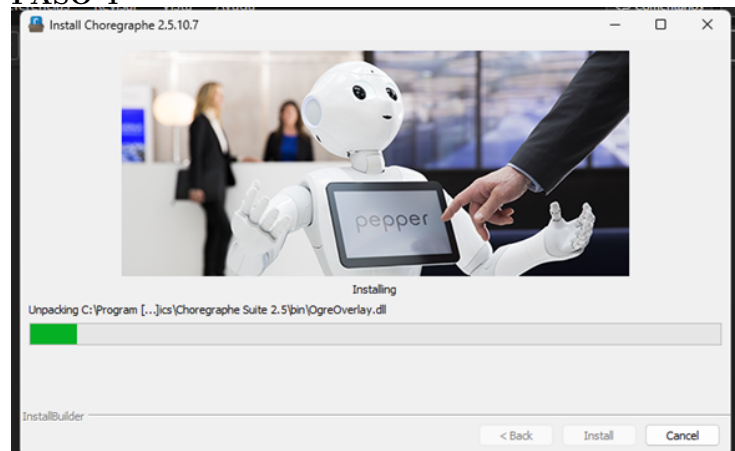
Aceptamos sus términos y le damos "Next".

PASO 3



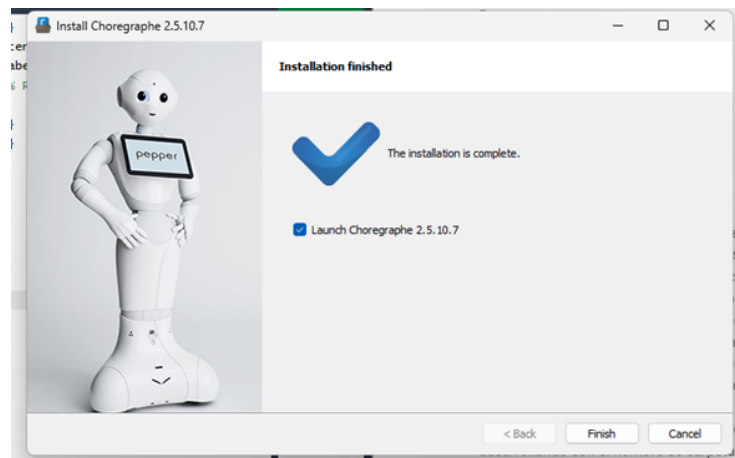
Le damos "Quick" "Next".

PASO 4



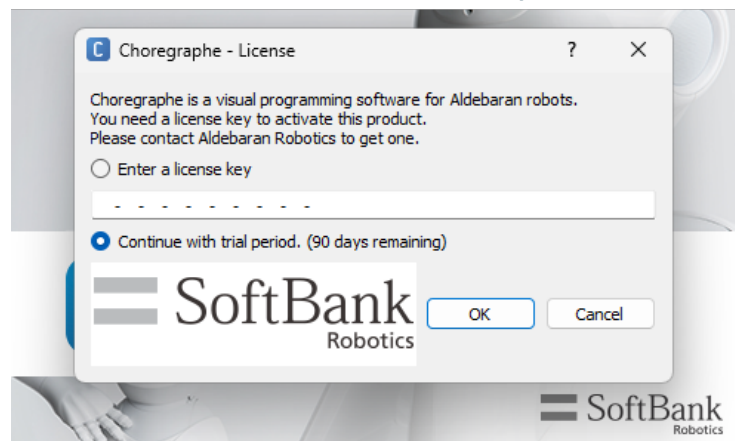
Le damos Install esperamos a que termine su instalación.

PASO 5



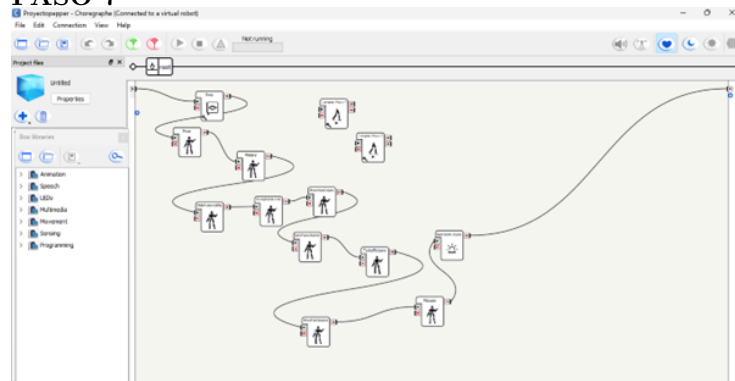
Le damos "Finish". Ahora tenemos nuestro programa instalado.

PASO 6



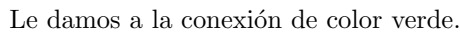
Le damos a "Continuar con un periodo de prueba por 90 días".

PASO 7

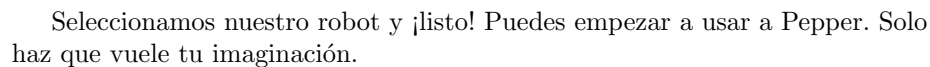


En la parte de abajo seleccionamos los movimientos, gestos, bailes que que-

PASO 8

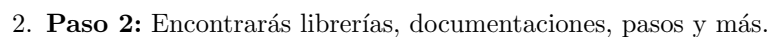


PASO 9



2. Pepper Terminal

1. **Paso 1:** Primero debes abrir la terminal desde Ubuntu colocar los datos de conectividad al robot (ssh, @, dirección IP) estando conectado en la red de "ciego".



- Paso 3:** Deberás seleccionar los bailes que tiene predeterminados y usarlos para hacer una coreografía, este archivo se encuentra desde "Bakend.py".

- Paso 4:** También puedes crear un archivo nano y colocar a través de las librerías explicadas y hacer movimientos con el, luego ejecutas tu archivo creado y listo.

3. Explicación de Código

Este código tiene como objetivo controlar a un robot Pepper para que realice una secuencia de baile de la "Macarena".

3.1. Estructura General

- Creación de proxies:** Se crea una serie de conexiones (proxies) con los servicios del robot (como el movimiento, la postura, la síntesis de voz, y el tablet) utilizando la IP y el puerto de Pepper.

2. **Preparación del robot:** Se manda al robot a la postura inicial (de pie) usando `goToPosture()`, y se hace que diga "Voy a bailar la macarena" usando `tts.say()`.
3. **Habilitación del movimiento:** Se activan los movimientos del robot (por ejemplo, los brazos) mediante el comando `motion.wakeUp()`, y se habilita el movimiento de los brazos a través de `motion.setMoveArmsEnabled(True, True)`.

3.2. Función macarena ciclo

La función principal que define los pasos del baile de la "Macarena". El robot realiza una serie de movimientos de brazo y cadera, simulando los pasos más conocidos de la danza. Estos movimientos se definen en forma de ángulos para cada una de las articulaciones de los brazos y la cadera del robot.

3.3. Secuencia de Movimientos

Cada paso (como levantar el brazo derecho o girar los brazos) está representado por un conjunto de ángulos que definen cómo deben moverse las articulaciones. Estos movimientos se realizan de forma secuencial, y cada uno tiene un tiempo de duración (`beat.time`).

El robot mueve los brazos hacia ciertas posiciones (por ejemplo, extender el brazo derecho, mover la mano a la cabeza, etc.).

Después de cada secuencia, se realiza una transición suave de un movimiento al siguiente usando el método `motion.angleInterpolation()`.

3.4. Pequeño giro ("¡Hey Macarena!")

Después de completar los pasos del baile, se realiza un pequeño giro o con el comando `motion.moveTo()`, mientras el robot dice "¡Hey Macarena!" usando `tts.say()`.

3.5. Ejecución del Baile

La función `macarena_dance_cycle()` se ejecuta tres veces (con el ciclo `for i in range(3)`) para que el robot repita el baile tres veces mientras la música y el video se reproducen en el fondo.

3.6. Manejo de Errores

Se utilizan bloques `try-except` para manejar posibles errores durante la ejecución, como problemas al reproducir el video o la música. Si ocurre un error, el robot se detiene y vuelve a la postura inicial.

Al final de la ejecución (en el bloque `finally`), se asegura que los movimientos del robot se detengan con `motion.stopMove()`, que se vuelve a la postura inicial.

con `posture.goToPosture("StandInit", 0.5)` y finalmente el robot se pone en descanso con `motion.rest()`.