

Laboratorio 2 — Interacción inicial con el Robot Pepper

Miguel Angel Montaña Sánchez

Jeferson Jair Hernández Garzón

Yossed Mauricio Riaño Páez

Curso: Sistemas Digitales 3 — Universidad Santo Tomás

Resumen

Este laboratorio tuvo como objetivo introducirnos en la programación y control del robot humanoide **Pepper** de SoftBank Robotics. Para esto, investigamos las librerías principales utilizadas en su funcionamiento, realizamos una coreografía sencilla con la herramienta **Choregraphe**, y probamos movimientos básicos mediante scripts en Python accediendo al robot a través de **SSH**. Finalmente, se documentó todo el proceso en Overleaf y se subió al repositorio de GitHub.

Introducción

La práctica se centró en la primera interacción con el robot **Pepper** en el salón de robótica. Con esta actividad aprendimos a usar las herramientas de desarrollo que permiten programar y controlar al robot, así como a ejecutar comandos y scripts por medio de la terminal. Además, exploramos cómo crear movimientos básicos y coreografías, lo que nos permitió comprender mejor las capacidades del robot.

Marco Teórico

Robot Pepper y su entorno

Pepper es un robot humanoide diseñado para interactuar de manera natural con las personas. Es capaz de reconocer emociones y responder en consecuencia. Su sistema operativo, **NAOqi**, controla sensores, actuadores y módulos de comportamiento.

Herramientas utilizadas

- **Choregraphe**: entorno de desarrollo gráfico que permite crear comportamientos arrastrando bloques y también añadir scripts en Python.
- **SSH (Secure Shell)**: protocolo para acceder de forma remota al sistema de Pepper y ejecutar comandos o scripts directamente.

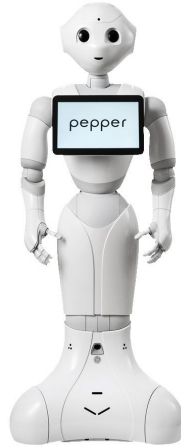


Figura 1: Robot Pepper

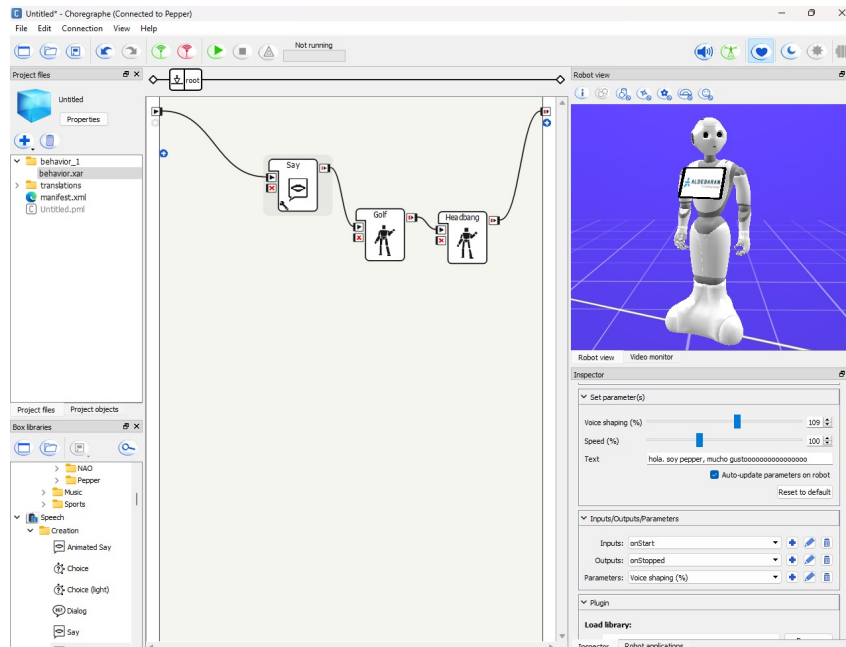


Figura 2: Entorno de programación Choregraphe

Librerías Utilizadas

Las siguientes librerías son fundamentales para el desarrollo de aplicaciones en Pepper:

- **qi**: Librería proporcionada por SoftBank Robotics que facilita la comunicación con los servicios de NAOqi, permitiendo controlar aspectos como movimiento, habla y percepción.
- **argparse**: Librería estándar de Python que permite manejar argumentos desde la línea de comandos, útil para parametrizar scripts.
- **sys**: Proporciona acceso a variables y funciones que interactúan con el intérprete de Python, como la gestión de argumentos y la salida del programa.

- **os**: Permite interactuar con el sistema operativo, gestionar archivos y directorios, y ejecutar comandos del sistema.
- **almath**: Librería matemática específica de SoftBank Robotics que proporciona herramientas para cálculos de transformaciones y cinemática, esenciales para el control de movimientos.
- **math**: Librería estándar de Python que ofrece funciones matemáticas básicas como trigonometría, logaritmos, etc.
- **motion**: Hace referencia al módulo ALMotion de NAOqi, responsable del control de movimientos del robot, incluyendo articulaciones y locomoción.
- **httpplib**: Librería estándar de Python 2 para manejar conexiones HTTP, utilizada para realizar solicitudes web.
- **json**: Permite trabajar con datos en formato JSON, facilitando la serialización y deserialización de información estructurada.

Librería qi

Descripción: Librería específica de SoftBank Robotics.

Uso principal en Pepper: Permite la comunicación con el framework NAOqi, que es el sistema operativo del robot. Facilita la conexión con los módulos del robot (cámara, audio, movimiento, etc.).

Funciones comunes: Creación de sesiones, carga de servicios, control de eventos, comunicación con otros dispositivos.

Librería argparse

Descripción: Librería estándar de Python.

Uso principal: Analiza argumentos pasados por línea de comandos al ejecutar scripts.

En Pepper: Permite que los scripts sean más configurables, como pasar la IP y puerto del robot al script.

Librería sys

Descripción: Librería estándar de Python.

Uso principal: Proporciona acceso a variables y funciones que interactúan con el intérprete de Python.

En Pepper: Se usa comúnmente para acceder a `sys.argv` o para terminar scripts con `sys.exit()`.

Librería os

Descripción: Librería estándar de Python.

Uso principal: Interacción con el sistema operativo, manejo de archivos y directorios.

En Pepper: Puede usarse para acceder a rutas de archivos, configuraciones, etc.

Librería `almath`

Descripción: Librería matemática específica de SoftBank Robotics.

Uso principal en Pepper: Realiza cálculos de cinemática, matrices de transformación, orientación y posición del robot.

Muy útil para: Manipulación avanzada del cuerpo del robot y navegación espacial.

Librería `math`

Descripción: Librería estándar de Python.

Uso principal: Funciones matemáticas básicas como seno, coseno, raíz cuadrada, etc.

En Pepper: Se usa junto con `almath` para cálculos más precisos en control de movimiento.

Librería `motion`

Descripción: Generalmente hace referencia al módulo `ALMotion` de `NAOqi`.

Uso principal en Pepper: Controla el movimiento del robot (articulaciones, equilibrio, desplazamiento).

Librería `httplib`

Descripción: Librería para hacer conexiones HTTP en versiones antiguas de Python (2.x).

Uso en Pepper: Permite comunicación web (como envío de datos o conexión con servidores).

Librería `json`

Descripción: Librería estándar de Python.

Uso principal: Codificación y decodificación de datos en formato JSON.

Uso en Pepper: Muy útil para leer archivos de configuración o enviar datos estructurados a través de la red.

Procedimiento y Desarrollo

Para la tarea 6 se solicitó desarrollar la instalación, descripción del paso a paso con **Choregraphe** y generar una secuencia sencilla de pasos por medio del aplicativo donde se creen movimientos básicos para el robot Pepper. A continuación se documenta el proceso:

Instalación de Choregraphe

- Durante la instalación se deben aceptar los términos y condiciones de uso del software.
- Luego se selecciona el modo de instalación adecuado para el equipo.

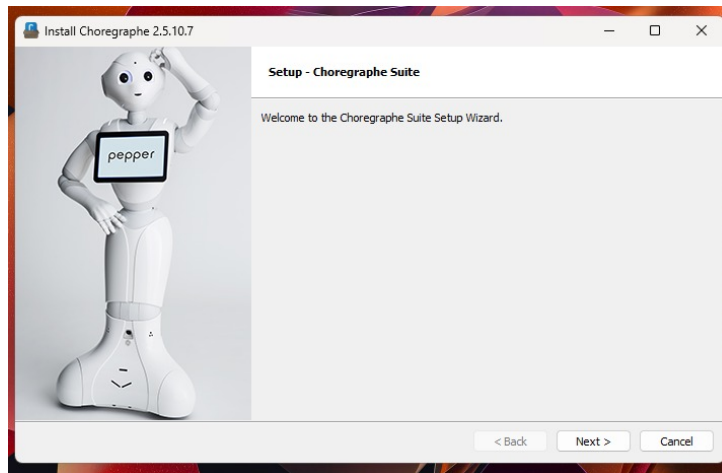


Figura 3: Comienza la instalación de Choregraphe

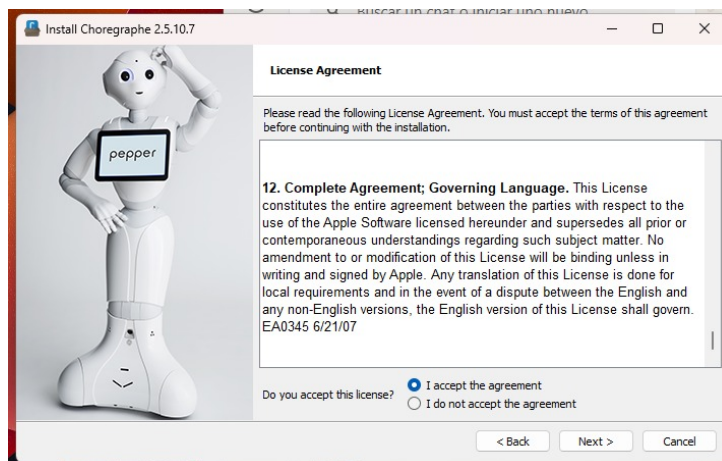


Figura 4: Aceptación de términos y condiciones

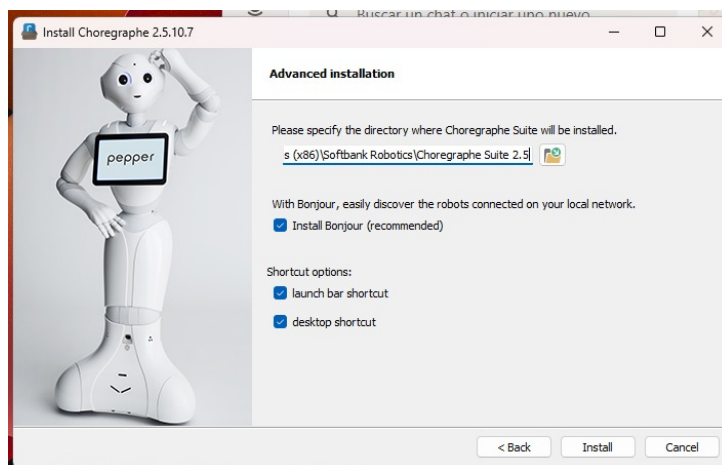


Figura 5: Selección del modo de instalación

Conexión de Choregraphe con Pepper

- Es necesario estar conectado a la misma red WiFi que utiliza Pepper.

- Al buscarlo aparecerán las opciones de robots disponibles según la red.
- Si no aparece, se puede ingresar el ID o la dirección IP directamente en la opción *Use fixed IP/hostname*.

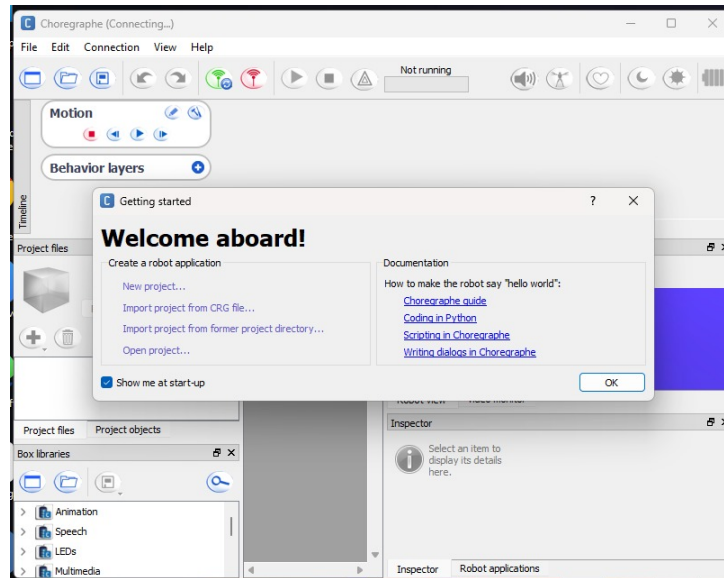


Figura 6: Conexión de Choregraphe a Pepper

Configuración de idioma

En nuestro caso fue necesario cambiar el idioma de inglés a español, ya que de esta forma el entorno podía usarse correctamente.

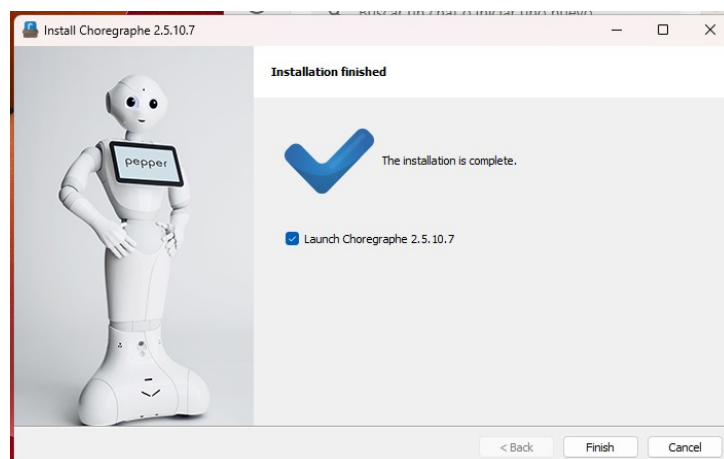


Figura 7: Cambio de idioma en la configuración de Choregraphe

Ejecución de movimientos

- En la parte inferior izquierda de la pantalla se encuentran carpetas con una variedad de movimientos e interacciones prediseñadas.

- Estos bloques se arrastran a la parte central y se conectan las entradas y salidas correspondientes.
- Para ejecutar, se verifica que el ícono del corazón en la parte superior derecha esté en azul (indicando conexión con Pepper).
- Una vez conectado, Pepper responde a las entradas programadas.

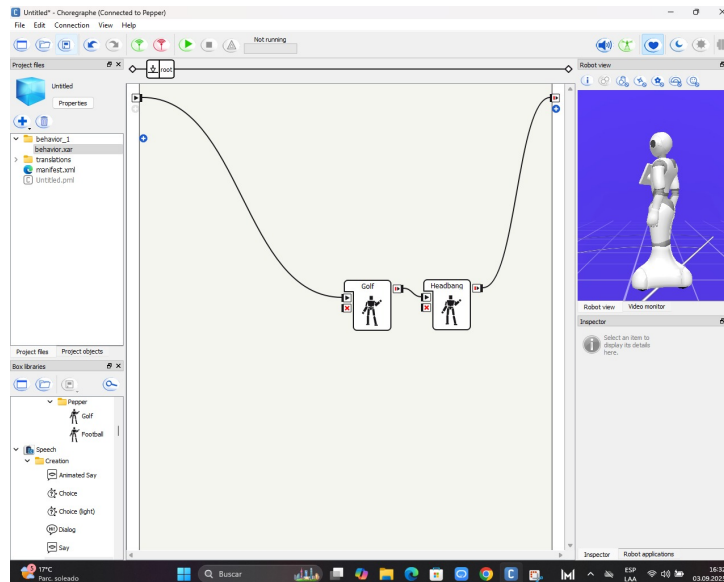


Figura 8: Ejecución de movimientos en Choregraphe

Creación de movimientos en Python desde consola

Además de Choregraphe, se realizaron pruebas creando movimientos en Python directamente desde la consola, accediendo al robot con:

```
ssh nao@192.168.0.100
```

Luego, editando un archivo en **nano** y ejecutándolo para que Pepper realizara movimientos simples, utilizando librerías como **qi** y **motion**.

Conclusiones

Este laboratorio nos permitió conocer de forma práctica cómo interactuar con el robot Pepper. El uso de Choregraphe fue útil para crear coreografías simples, mientras que la conexión SSH nos dio la posibilidad de escribir y ejecutar scripts en Python, logrando movimientos más personalizados. Además, el estudio de las librerías nos ayudó a entender mejor cómo se estructura la programación en Pepper y su potencial en aplicaciones más complejas.

```

miguel@Miguel:~/Escritorio$ ssh nao@192.168.0.104>
bash: error sintáctico cerca del elemento inesperado `newline'
miguel@Miguel:~/Escritorio$ ssh nap@192.168.0.104>
bash: error sintáctico cerca del elemento inesperado `newline'
miguel@Miguel:~/Escritorio$ ssh nao@192.168.0.104
The authenticity of host '192.168.0.104 (192.168.0.104)' can't be established.
ED25519 key fingerprint is SHA256:AQvG0R1zozJyyH8epTcbVK+YusdNSLamclA5nV6U0L4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.0.104' (ED25519) to the list of known hosts.
(nao@192.168.0.104) Password:
Pepper [0] ~ $ whoami uname -a python2.7 --version qicli services
whoami: invalid option -- 'a'
Try 'whoami --help' for more information.
Pepper [err 1] ~ $ mkdir -p ~/proyectos/pepper
Pepper [0] ~ $ cd ~/proyectos/pepper
Pepper [0] ~/proyectos/pepper $ cd ~/proyectos/pepper
Pepper [0] ~/proyectos/pepper $ nano hola_pepper.py
Pepper [0] ~/proyectos/pepper $ chmod +x hola_pepper.py
Pepper [0] ~/proyectos/pepper $ python2.7 hola_pepper.py --ip 127.0.0.1 --port 9
559

```

Figura 9: Código en Python para movimiento básico de Pepper

```

Pepper [0] ~/proyectos/pepper $ cd ~/proyectos/pepper
Pepper [0] ~/proyectos/pepper $ nano hola_pepper.py
Pepper [0] ~/proyectos/pepper $ chmod +x hola_pepper.py
Pepper [0] ~/proyectos/pepper $ python2.7 hola_pepper.py --ip 127.0.0.1 --port 9
559
[W] 1756936090.558166 3688 qi.path.sdklayout: No Application was created, trying
to deduce paths
90 grados son 1.5708 rad
Pepper [0] ~/proyectos/pepper $ python2.7 hola_pepper.py --ip 127.0.0.1 --port 9
559
[W] 1756936126.491421 3699 qi.path.sdklayout: No Application was created, trying
to deduce paths
90 grados son 1.5708 rad
Pepper [0] ~/proyectos/pepper $ python2.7 hola_pepper.py --ip 192.168.0.104 --po
rt 9559
[W] 1756936411.700077 3730 qi.path.sdklayout: No Application was created, trying
to deduce paths
90 grados son 1.5708 rad
Pepper [0] ~/proyectos/pepper $ python2.7 hola_pepper.py --ip 192.168.0.104 --po
rt 9559
[W] 1756936514.574115 3743 qi.path.sdklayout: No Application was created, trying
to deduce paths
90 grados son 1.5708 rad

```

Figura 10: Código en Python para movimiento básico de Pepper