

Informe del Proyecto: Presentación con Pepper y Dashboard Tecnológico

Miguel Caro

Samuel Parra

2025

Índice general

1. Introducción	2
2. Desarrollo del Proyecto	3
2.1. Creación de diapositivas	3
2.2. Código en Python para la exposición con Pepper	3
2.3. Evidencia de la exposición	4
3. Desarrollo del Dashboard con Streamlit	9
3.1. Preparación del entorno	9
3.2. Estructura del Dashboard	9
3.3. Ejemplo de implementación	10
3.4. Ejecución del Dashboard	10
3.5. Resultados	10
3.6. Manejo de GitHub y Readme	11
4. Conclusiones	12

Capítulo 1

Introducción

El presente informe describe el proceso de creación de un proyecto académico en el cual se integraron diferentes tecnologías para realizar una exposición con el robot Pepper.

El trabajo se desarrolló en tres fases principales:

- Creación de diapositivas y preparación del guion.
- Programación en Python para que Pepper realizara la exposición.
- Desarrollo de un Dashboard con Streamlit que integra chatbot y videos.

Capítulo 2

Desarrollo del Proyecto

2.1. Creación de diapositivas

La primera parte consistió en diseñar las diapositivas de la exposición. Para ello se utilizó la herramienta en línea **Gamma**, que permite crear presentaciones dinámicas de manera sencilla.

En esta fase también se realizó la investigación sobre los temas a exponer: minería de asteroides, neuroprótesis inteligentes e interfaces cerebro-computador. Con esta información se preparó el guion que más adelante sería interpretado por Pepper.

2.2. Código en Python para la exposición con Pepper

Una vez listas las diapositivas y el guion, se procedió a programar al robot Pepper. Para ello se desarrolló un script en Python que sincroniza:

- La visualización de cada diapositiva en la tablet del robot.
- Los movimientos de los brazos.
- El discurso hablado mediante el servicio de *ALAnimatedSpeech*.

El siguiente fragmento muestra un ejemplo del código implementado:

```
st.title("Novedades Tecnológicas con Pepper")

# Video de inicio
st.subheader("Video de Presentación")
st.video("https://www.youtube.com/watch?v=tu_link")
```

```
# Novedades Tecnológicas
st.header("Minería de Asteroides")
st.write("Breve descripción de esta tecnología...")

st.header("Neuroprótesis Inteligentes")
st.write("Breve descripción de esta tecnología...")

st.header("Interfaces Cerebro-Computador")
st.write("Breve descripción de esta tecnología...")

# Chatbot simulado
st.subheader("Chatbot de confianza")
pregunta = st.text_input("Escribe tu pregunta:")
if pregunta:
    st.write(f"Respuesta simulada: Gracias por preguntar sobre {pregunta}")
```

2.3. Evidencia de la exposición

Las siguientes imágenes corresponden a las diapositivas que Pepper mostró durante la exposición, acompañadas de narración y gestos programados en Python.



Figura 2.1: Portada de la exposición.



Figura 2.2: Introducción: Desafíos y Avances Disruptivos.



Figura 2.3: Minería de Asteroides: La Nueva Frontera de Recursos.



Figura 2.4: Neuroprótesis Inteligentes: Restaurando Funciones Humanas.



Figura 2.5: El Software Libre en Salud: Un Catalizador de Innovación.



Figura 2.6: Interfaces Cerebro-Computador: Uniendo Mente y Máquina.



Figura 2.7: Sistemas Digitales en BCI: La Precisión en Tiempo Real.

Conclusión: Un Futuro de Posibilidades Infinitas

1	2	3
Recursos Sin Límites La minería de asteroides nos abre la puerta a una abundancia de recursos más allá de la Tierra, impulsando la exploración y la sostenibilidad a largo plazo.	Salud Accesible Las neuroprótesis inteligentes, potenciadas por el software libre, prometen una salud más innovadora, personalizada y accesible para todos.	Mente y Tecnología Las BCI no invasivas representan la unión más directa entre la mente humana y el mundo digital, desbloqueando nuevas formas de interacción y capacidad.

Estos tres ejemplos son solo el comienzo de cómo la ciencia y la tecnología están abriendo **nuevas fronteras** para la humanidad, forjando un futuro donde lo que antes era ciencia ficción se convierte en nuestra realidad.

Made with **GAMMA**

Figura 2.8: Conclusión: Un Futuro de Posibilidades Infinitas.

Capítulo 3

Desarrollo del Dashboard con Streamlit

La última fase del proyecto consistió en la creación de un **Dashboard interactivo** con *Streamlit*, cuyo objetivo fue integrar en una sola interfaz tanto el video de la exposición de Pepper como un chatbot simulado para responder preguntas sobre la exposición.

3.1. Preparación del entorno

Para comenzar se instaló la librería Streamlit en el entorno de Python mediante el comando:

```
pip install streamlit
```

Posteriormente, se creó el archivo `DASHBOARD.py` dentro de la carpeta del proyecto. Este archivo contiene toda la estructura de la aplicación web.

3.2. Estructura del Dashboard

El diseño del tablero se organizó en cuatro bloques principales:

1. **Título y presentación general:** se muestra el encabezado de la aplicación con el nombre del proyecto.
2. **Video de inicio:** se integró el video de la exposición de Pepper, permitiendo reproducirlo directamente desde la interfaz.

3. **Temas de la Exposición:** cada tema investigado (Minería de Asteroides, Neuroprótesis Inteligentes e Interfaces Cerebro-Computador) cuenta con un apartado que incluye un breve resumen.
4. **Chatbot:** se añadió una caja de texto donde el usuario puede escribir preguntas y recibir respuestas simples preprogramadas, demostrando el concepto de interacción hombre-máquina.

3.3. Ejemplo de implementación

El siguiente fragmento de código muestra cómo se integró el video de presentación y una de las secciones de novedades tecnológicas:

```
import streamlit as st

st.title("Novedades Tecnológicas con Pepper")

# Video de inicio
st.subheader("Video de Presentación")
st.video("https://www.youtube.com/watch?v=tu_link")

# Novedad tecnológica 1
st.header("Minería de Asteroides")
st.write("La minería espacial busca extraer metales preciosos y agua de asteroides, reduciendo costos de exploración y abriendo nuevas fronteras de recursos.")
```

3.4. Ejecución del Dashboard

Para ejecutar el Dashboard se utilizó el siguiente comando en consola:

```
streamlit run DASHBOARD.py
```

Esto permitió desplegar el tablero en un navegador web (por defecto en <http://localhost:8501>), ofreciendo una interfaz sencilla, clara y funcional.

3.5. Resultados

El Dashboard final permitió centralizar la información del proyecto en un solo lugar. Los usuarios pueden visualizar el video de Pepper, leer las

descripciones de las innovaciones tecnológicas y realizar consultas al chatbot simulado. De esta manera, se integraron tanto la programación robótica como el desarrollo de interfaces digitales en un mismo producto académico.

3.6. Manejo de GitHub y Readme

Para unificar ramas en GitHub, se accedió al repositorio, se seleccionó "Pull requests" luego "New pull request", donde se designó la rama base como destino (main) y la rama compare con los cambios a integrar; tras revisar las modificaciones, se creó el pull request completando la información requerida y finalmente se ejecutó la unificación mediante merge tradicional, squash merge o rebase merge según correspondiera.

Capítulo 4

Conclusiones

El proyecto permitió integrar diferentes herramientas digitales y de programación, aplicadas al uso del robot Pepper:

- Se diseñaron diapositivas interactivas en Gamma.
- Se programó un guion completo para la exposición en Python.
- Se construyó un Dashboard en Streamlit que permite mostrar videos y un chatbot simulado.

Este trabajo favoreció el aprendizaje en programación, control de robots, uso de repositorios GitHub y despliegue de aplicaciones interactivas.