

Proyecto Integrado con Pepper, Chatbot y Dashboard

Juan Pablo Pedraza - Marco Antonio Gómez

8 de septiembre de 2025

Resumen

Este documento describe el desarrollo de un proyecto integrado que combina la interacción con el robot Pepper, el desarrollo de un chatbot personalizado, la creación de un dashboard en Streamlit, la documentación en Overleaf y el manejo de GitHub. Se explican paso a paso las actividades realizadas en cada punto, con evidencias gráficas y explicaciones detalladas.

Índice

1. Primer Punto: Interactuando con Pepper	2
1.1. Temas Seleccionados	2
1.2. Desarrollo	2
1.3. Evidencias	2
2. Segundo Punto: Desarrollando Chatbot Personalizado	3
2.1. Objetivo	3
2.2. Desarrollo	3
2.3. Evidencias	3
3. Tercer Punto: Desarrollando Dashboard Integrado	4
3.1. Tecnología Usada: Streamlit	4
3.2. Desarrollo	4
3.3. Evidencias	4

1. Primer Punto: Interactuando con Pepper

En este apartado se describe la interacción con Pepper y la exposición de 3 a 5 minutos sobre tres temas tecnológicos seleccionados.

1.1. Temas Seleccionados

- Tema 1: Impacto de los sistemas digitales en implantes cerebrales adaptativos.
- Tema 2: Biocombustibles de algas genéticamente editadas.
- Tema 3: Computación reversible y su impacto en los sistemas digitales.

1.2. Desarrollo

- Creación de carpetas y archivos del proyecto.
- Programación de movimientos y alocución en Pepper.
- Ensayos y pruebas de exposición.

1.3. Evidencias

Aquí se insertan capturas de pantalla del código y fotos del robot Pepper ejecutando la exposición.

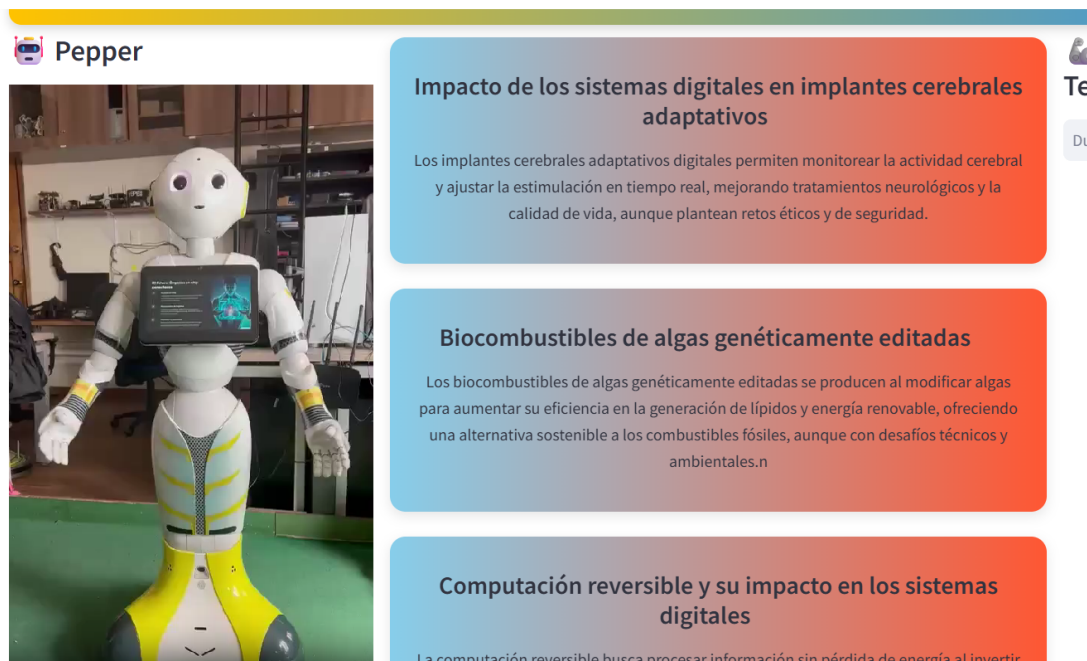


Figura 1: Evidencia 1

2. Segundo Punto: Desarrollando Chatbot Personalizado

2.1. Objetivo

Construir un chatbot especializado en los temas seleccionados.

2.2. Desarrollo

1. Creación de carpeta y estructura del chatbot.
2. Instalación de librerías necesarias en consola.
3. Desarrollo de respuestas automáticas (texto y voz).
4. Pruebas de interacción con Pepper.

2.3. Evidencias

- código para que muestre imagen en la tablet:

```
class MyClass(GeneratedClass):
    def _init_(self):
        GeneratedClass._init_(self)
        self.tablet = None

    def onLoad(self):
        try:
            self.tablet = self.session().service("ALTabletService")
            self.logger.info("Conectado a ALTabletService")
        except Exception as e:
            self.logger.error("No pude conectar con ALTabletService: " + str(e))

    def onInput_onStart(self):
        # Prueba 1: imagen pública en Internet (debe ser URL directa)
        url = "https://raw.githubusercontent.com/usuario/repo/branch/path/imagen.png"
        # Prueba 2: imagen dentro del paquete (descomenta si usas esta opción)
        # url = "http://198.18.0.1/apps/miPaquete/html/imagen.png"

        try:
            self.logger.info("Intentando mostrar imagen: " + url)
            self.tablet.showImage(url)
        except Exception as e:
            self.logger.error("showImage fallo: {}".format(e))
            # fallback: intentar abrirla en webview
            try:
                self.tablet.showWebview(url)
                self.logger.info("Fallback: showWebview ok")
            except Exception as e2:
```

```

        self.logger.error("showWebview fallo: {}".format(e2))

def onInput_onStop(self):
    pass

```

3. Tercer Punto: Desarrollando Dashboard Integrado

3.1. Tecnología Usada: Streamlit

Se explica cómo se construyó el dashboard integrando el chatbot y los videos de Pepper.

3.2. Desarrollo

- Creación del archivo `app.py`.
- Integración del chatbot en la interfaz.
- Integración de videos y resultados de Pepper.
- Ejecución con el comando `streamlit run app.py`.

3.3. Evidencias

- Código para que cree el Dashboard:

```

# app.py
import streamlit as st
import os
import chatbot_component
#         tu chatbot est     en este archivo separado

st.set_page_config(page_title="Pepper Dashboard", layout="wide", initial_sidebar_state="collapsed")

# -----
# CSS
# -----
st.markdown(
    """
    <style>
    .header {
        background: linear-gradient(90deg, #FFC300, #1E90FF);
        padding: 14px;
        border-radius: 12px;
        text-align: center;
        font-weight: 700;
        font-size: 22px;
        color: black;
        box-shadow: 0px 4px 10px rgba(0,0,0,0.15);
    }
    """
)

```

```

        .card {
            background: linear-gradient(90deg, #87CEEB, #FF5733);;
            padding: 20px;
            border-radius: 15px;
            text-align: center;
            margin: 12px 0;
            box-shadow: 0px 4px 12px rgba(0,0,0,0.1);
        }
    </style>
    """
    unsafe_allow_html=True
)

# -----
# Header
# -----
st.markdown("<div class='header'>UN POCO DE NOVEDADES TECNOLGICAS</div>")

# -----
# Layout
# -----
col1, col2, col3 = st.columns([0.9, 1.6, 0.9])

# -----
# Columna 1: Video
with col1:
    st.subheader("Pepper")
    video_path = "pepper_video.mp4"
    if os.path.exists(video_path):
        st.video(video_path) # siempre aparece
    else:
        st.warning("No encontr el archivo 'pepper_video.mp4'. S b")

# -----
# Columna 2: Novedades
# -----
with col2:
    st.markdown("<div class='card'><h4>Impacto de los sistemas digitales en")
    st.markdown("<div class='card'><h4> Biocombustibles de algas gen ticam")
    st.markdown("<div class='card'><h4>Computaci n reversible y su impacto")

# -----
# Columna 3: Tu Chatbot
# -----
with col3:
    chatbot_component.show_chatbot() # aqu se ejecuta tu chatbot

```

1. Evidencia de Pepper interactuando y exponiendo los temas asignados: [Video de Pepper](#)
2. Evidencia del dashboar completo: [Dashboard de Pepper](#)