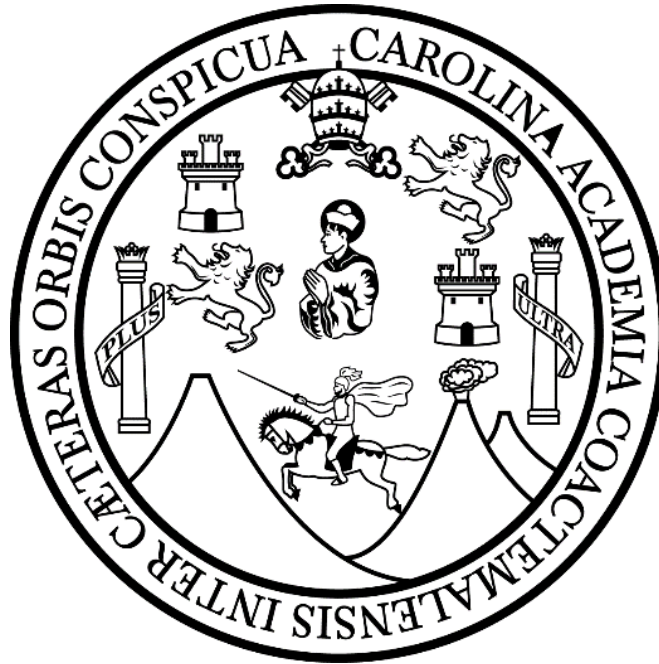


UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE VACACIONES SEGUNDO SEMESTRE 2024



MANUAL TÉCNICO FASE 1
LABORATORIO DE INTELIGENCIA ARTIFICIAL 1
GRUPO #1
AUX. JOSÉ ANDRÉS MONTENEGRO SANTOS

MARIANO FRANCISCO CAMPOSECO CAMPOSECO	202030987
LUIS NERY CIFUENTES RODAS	202030482
MANUEL ANTONIO ROJAS PAXTOR	202030799

10 DE DICIEMBRE DE 2024

Configuración Local

Requisitos de Hardware

1. Servidor:

- CPU: Mínimo de 2 núcleos (se recomienda 4 núcleos o más para un rendimiento óptimo).
- RAM: Mínimo de 4 GB (se recomienda 8 GB o más para manejar múltiples conexiones simultáneas).
- Almacenamiento: Mínimo de 20 GB de espacio disponible (se recomienda SSD para mejor rendimiento).

2. Dispositivos de Cliente:

- Navegador moderno y actualizado (Chrome, Firefox, Edge, Safari) con soporte para HTML5 y CSS3.

Requisitos de Software

1. Sistema Operativo:

- Linux (Ubuntu, CentOS, Debian) o Windows Server 2016 y versiones superiores.

2. Node.js:

- Versión: 16.x o superior (se recomienda la última LTS).
- npm: Versión 8.x o superior (instalado automáticamente con Node.js).

3. Angular:

- Angular CLI: Versión 14.x o superior (se puede instalar globalmente con npm).
- Dependencias adicionales: Asegúrate de instalar todas las dependencias necesarias según el archivo package.json.

4. Servidor Web:

- Nginx o Apache (se recomienda Nginx por su eficiencia en la gestión de conexiones concurrentes).

5. Otras Dependencias:

- cURL.
- OpenSSL.
- Git (para gestionar el control de versiones y clonar repositorios).

Instalación de Requisitos de Software para RunesTalk

1. Sistema Operativo

Se recomienda utilizar la distribución de **Ubuntu o Debian**. Para la instalación en Linux, asegúrate de tener privilegios de superusuario.

2. Node.js

Instalación:

Para **Ubuntu**:

- `sudo apt update`
- `sudo apt install -y curl`
- `curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -`
- `sudo apt install -y nodejs`

Verificación:

Comprueba la versión instalada:

- `node -v`
- `npm -v`

3. Angular

Instalación de Angular CLI:

Ejecuta el siguiente comando para instalar Angular CLI globalmente:

- `npm install -g @angular/cli`

Verificación:

Comprueba la versión instalada de Angular CLI:

- `ng version`

4. Dependencias Adicionales

Asegúrate de instalar todas las dependencias necesarias especificadas en el archivo `package.json` de tu proyecto. Desde el directorio raíz del proyecto, ejecuta:

- `npm install`

5. Servidor Web

Nginx (recomendado por su eficiencia):

Para **Ubuntu**:

- sudo apt update
- sudo apt install -y nginx

Verificación:

Comprueba que Nginx está corriendo:

- systemctl status nginx

6. Otras Dependencias

cURL:

Para **Ubuntu**:

- sudo apt install -y curl

OpenSSL:

Para **Ubuntu**:

- sudo apt install -y openssl

Git:

Para **Ubuntu**:

- sudo apt install -y git

Configuración para despliegue automático a GitHub Pages

Se necesita agregar las ramas que se utilizarán asimismo se agrega la carpeta dist para el despliegue ya que se encuentra utilizando angular el entorno, por lo tanto va de la par con npm, para que antes de intentar generar obtenga las dependencias necesarias para su funcionamiento.

```
name: Deploy runes talk

on:

  push:

    branches:

      - main

jobs:

  deploy:

    runs-on: ubuntu-latest

    steps:

      - name: Checkout repository

        uses: actions/checkout@v3

      - name: Setup Node.js

        uses: actions/setup-node@v3

        with:

          node-version: 22.3.0

      - name: Install dependencies

        working-directory: runes-talk
```

```

run: npm install

- name: Build Angular app

  working-directory: runes-talk

  run: npm run build -- --configuration production
  --output-path=../dist --base-href "/IA1_Proyecto_1/"

- name: Deploy to GitHub Pages

  uses: peaceiris/actions-gh-pages@v3

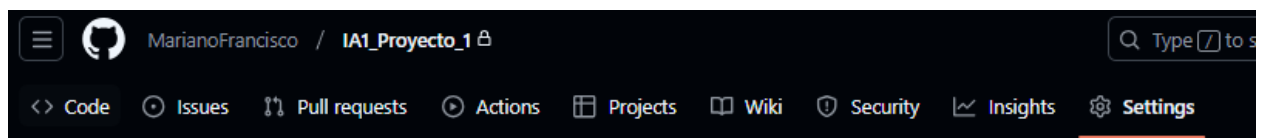
  with:

    github_token: ${ secrets.GITHUB_TOKEN }

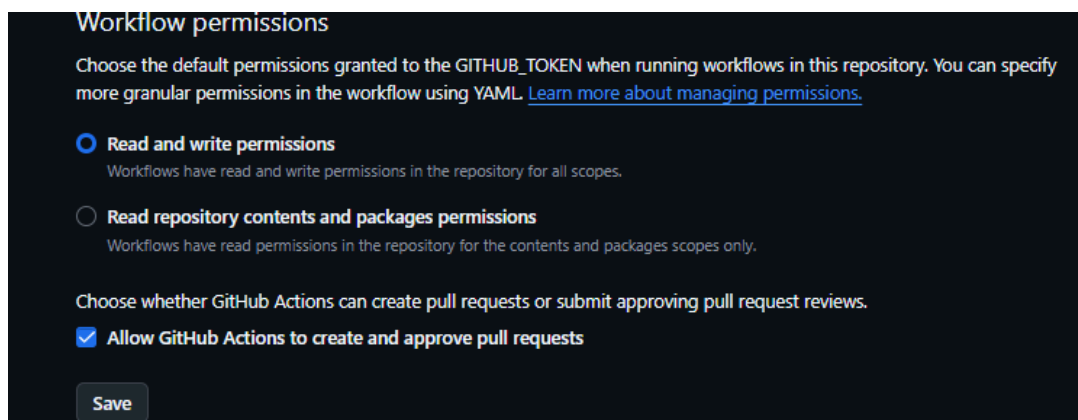
    publish_dir: dist/browser

```

Ir a settings



Activar Read and write permissions y Allow GitHub Actions to create and approve pull requests



Uso del modelo

@tensorflow/tfjs (instalar la dependencia)

npm install @tensorflow/tfjs

Sequential Model

Estructura de modelo en aprendizaje profundo (deep learning) proporcionada por bibliotecas como Keras, diseñada para construir redes neuronales de manera sencilla y secuencial.

Uso en Angular

Primeng

Biblioteca de componentes de interfaz de usuario (UI) para Angular, que proporciona una colección rica y personalizable de elementos para construir aplicaciones web modernas. Incluye componentes como botones, tablas, calendarios, gráficos, formularios, menús, entre otros, con temas predefinidos y soporte para estilos responsivos.

PrimeFlex

Biblioteca de utilidades CSS desarrollada por el equipo de PrimeTek que se utiliza para crear diseños responsivos y flexibles. Se basa en el modelo **Flexbox** y **CSS Grid**, proporcionando clases para espaciado, alineación, distribución y otros estilos comunes.

Estructura del proyecto

.git

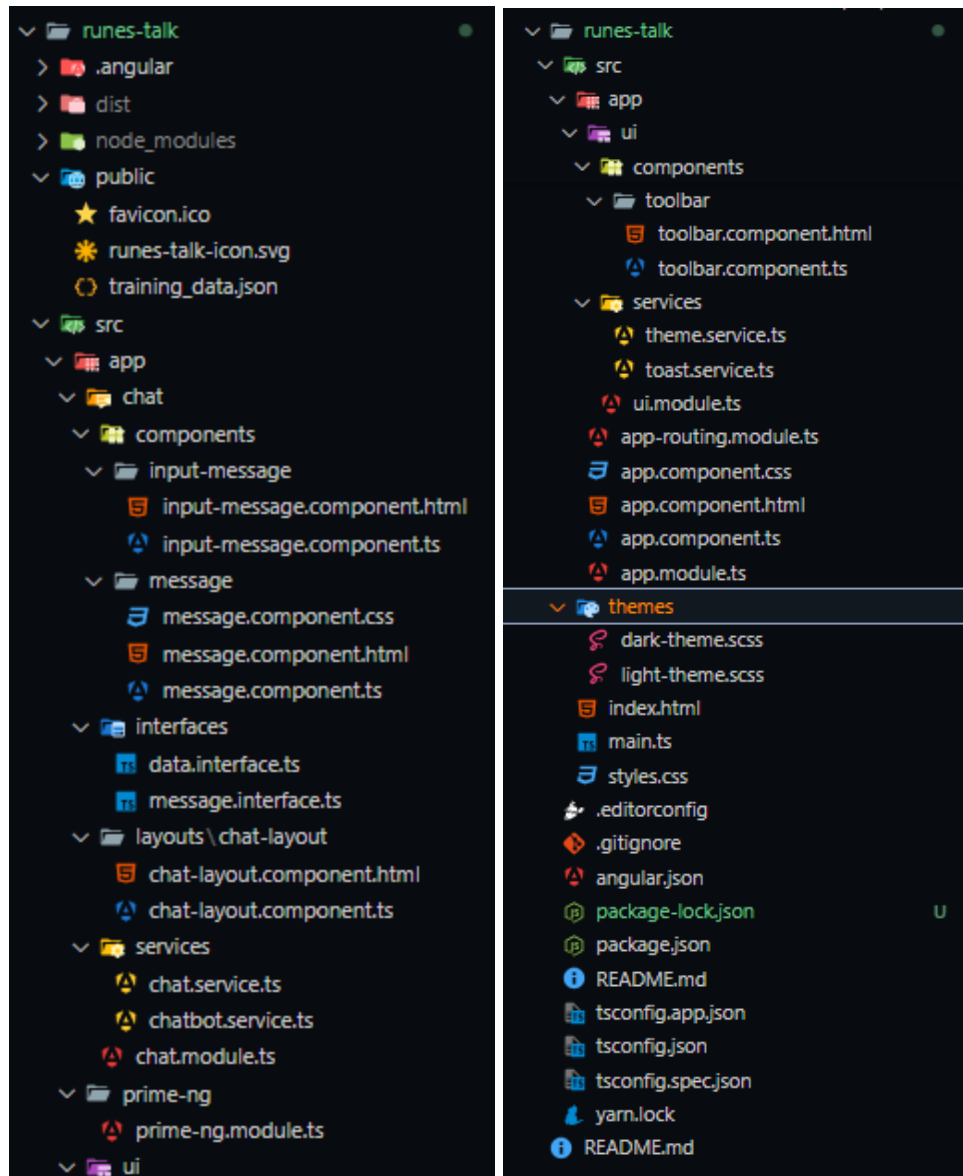
Directorio de control de versiones que almacena la configuración y el historial de cambios del repositorio. Facilita la gestión de versiones y el trabajo colaborativo entre desarrolladores.

.github/workflows

Directorio para almacenar archivos de despliegue

runes-talk

El directorio **runes-talk** se encarga de la lógica y presentación de la interfaz de usuario utilizando Angular, organizado de la siguiente manera:



- **.angular:** Directorio que almacena cachés y archivos temporales generados por Angular CLI.
- **public:** Archivos estáticos, como imágenes y el favicon de la aplicación.
- **src:** Contiene el código fuente de la aplicación Angular, subdividido en:
 - **app:** Módulos y componentes que implementan la lógica de la aplicación.
 - **theme:** Carpeta para temas, claro y oscuro.
- **angular.json:** Archivo de configuración del proyecto Angular, que define cómo se construye y sirve la aplicación.
- **package.json:** Archivo de configuración que especifica las dependencias y scripts del proyecto Angular.

Descripción de Componentes Clave

Frontend

- **Componentes:** Encapsulan la lógica y presentación de partes de la interfaz de usuario, permitiendo su reutilización.
- **Servicios:** Centralizan la lógica de acceso a datos y negocio, gestionando la comunicación con el backend.
- **Rutas:** Configuran la navegación dentro de la aplicación, permitiendo a los usuarios acceder a diferentes vistas.
- **Interfaces:** Definen la estructura de los datos utilizados en la aplicación, garantizando consistencia y tipado estricto.

Documento de despliegue

Documento de despliegue						
Número de Ticket: 0000000001						
Consultor		Consultor - Auxiliar				
Desarrollador		Desarrollador - mariano1941@outlook.es				
Riesgo Técnico	Alto		Medio		Bajo	X
Premisas o consideraciones para el pase a producción						
Fecha sugerida y hora sugerida para transportar el cambio		Se recomienda pasar este ticket a producción el fin de semana, para que los usuarios puedan empezar a utilizarlo el lunes.				
Configuraciones propias de ambientes u otros		Se necesita que cada característica esté en un servidor independiente, así mismo que se cuente con backups para la mejor seguridad, al ser el primer despliegue se requiere la prueba de toda la funcionalidad.				
Dependencia con otro ticket (si aplica)		No aplica				
Acciones a tomar en cuenta por si ocurre un error en el despliegue a QA y la solución no está completa		No aplica al ser el primer despliegue				
Pre-Requisitos		Acceso al entorno de Prod de RunesTalk como usuario administrador				
Recursos y Materiales						

Número		Detalles
1		Generación de logo a) IA
Secuencia de despliegue		
Número	Descripción del paso	
1	Envio a produccion a) Hacer git checkout a master b) Realizar git push	