

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

PLATAFORMA PARA DESARROLLO Y EVALUACIÓN DE
MODELOS DE RECONOCIMIENTO DE 11 GESTOS DE LA MANO

DESARROLLO DE UN SISTEMA PARA LA EVALUACIÓN Y
VISUALIZACIÓN DE LOS RESULTADOS DE MODELOS DE
RECONOCIMIENTO DE 11 GESTOS DE LA MANO

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE
SOFTWARE

KEVIN DAVID RAMOS ACARO

kevin.ramos01@epn.edu.ec

DIRECTOR: LORENA ISABEL BARONA LÓPEZ

lorena.barona@epn.edu.ec

DMQ, agosto 2023

CERTIFICACIONES

Yo, Kevin Ramos declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

KEVIN DAVID RAMOS ACARO

Certifico que el presente trabajo de integración curricular fue desarrollado por Kevin Ramos, bajo mi supervisión.

LORENA ISABEL BARONA LÓPEZ
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

KEVIN DAVID RAMOS ACARO

LORENA ISABEL BARONA LÓPEZ

DEDICATORIA

Dedico este Trabajo de integración curricular a mi familia y a todas las personas que estuvieron presentes en mi proceso de formación académica.

AGRADECIMIENTO

Agradezco a todos los profesores que aportaron en mi formación académica que facilitaron el desarrollo del proyecto de titulación con sus valiosos conocimientos. Además, a la institución Escuela Politécnica Nacional que me permitió desarrollarme profesionalmente.

ÍNDICE DE CONTENIDO

| | |
|---|------|
| CERTIFICACIONES..... | I |
| DECLARACIÓN DE AUTORÍA..... | II |
| DEDICATORIA..... | III |
| AGRADECIMIENTO..... | IV |
| ÍNDICE DE CONTENIDO..... | V |
| LISTA DE FIGURAS..... | VI |
| LISTA DE TABLAS..... | VII |
| LISTA DE ANEXOS | VII |
| RESUMEN | VIII |
| ABSTRACT | IX |
| 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO..... | 1 |
| 1.1 Objetivo general..... | 1 |
| 1.2 Objetivos específicos | 1 |
| 1.3 Alcance | 2 |
| 1.4 Marco teórico..... | 4 |
| 2 METODOLOGÍA..... | 6 |
| 2.1 Selección metodología modelo en cascada..... | 6 |
| 2.2 Definición JSON entrada..... | 6 |
| 2.3 Definición de requerimientos..... | 8 |
| 2.4 Diseño | 9 |
| 2.5 Implementación..... | 12 |
| 2.6 Integración y pruebas del sistema..... | 13 |
| 2.7 Operación..... | 15 |
| 2.8 Evaluación con el modelo de reconocimiento de gestos..... | 18 |
| 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES..... | 18 |
| 3.1 Resultados | 18 |
| 3.2 Conclusiones..... | 23 |
| 3.3 Recomendaciones..... | 24 |
| 4 REFERENCIAS BIBLIOGRÁFICAS..... | 25 |
| 5 ANEXOS..... | 26 |
| ANEXO I Casos de uso del sistema de evaluación..... | 27 |
| ANEXO II Diagrama de clases del sistema de evaluación | 31 |
| ANEXO III Manual de usuario del sistema de evaluación | 32 |

| | |
|--|----|
| ANEXO IV Código fuente del sistema de evaluación | 32 |
| ANEXO IV Video del funcionamiento del sistema de valuación con la integración del primer componente..... | 32 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 Arquitectura Cliente-Servidor | 4 |
| Figura 2 Arquitectura cliente-servidor del Sistema de Evaluación | 9 |
| Figura 3 Clase Evaluator..... | 10 |
| Figura 4 Clase Evaluated..... | 10 |
| Figura 5 Patrón de diseño "Factory" | 11 |
| Figura 6 Diagrama de secuencias evaluación de un modelo | 12 |
| Figura 7 Levantamiento servidor local Backend | 15 |
| Figura 8 Levantamiento servidor local Frontend | 16 |
| Figura 9 Pantalla principal sistema de evaluación HGR (Hand Gesture Recognition) | 16 |
| Figura 10 Página web sin servicio..... | 17 |
| Figura 11 Proceso de compilación y levantamiento fase de producción..... | 17 |
| Figura 12 Integración del primer y segundo componente..... | 17 |
| Figura 13 Pantalla Submit con modelo cargado..... | 21 |
| Figura 14 Proceso de evaluación del modelo..... | 21 |
| Figura 15 Resultados resumidos de la evaluación del modelo | 22 |
| Figura 16 Resultados de clasificación del modelo evaluado | 22 |
| Figura 17 Resultados de reconocimiento del modelo evaluado | 23 |
| Figura 18 Resultados de tiempos del modelo evaluado | 23 |

LISTA DE TABLAS

| | |
|---|----|
| Tabla 1 Gestos de la mano | 2 |
| Tabla 2 Caso de uso CU_01 | 8 |
| Tabla 3 Casos de prueba..... | 13 |
| Tabla 4 Pruebas de aceptación..... | 14 |
| Tabla 5 Resultados de las pruebas de integración..... | 19 |
| Tabla 6 Resultados obtenidos pruebas de aceptación | 19 |
| Tabla 7 Resultados obtenidos pruebas de aceptación con el primer componente | 20 |
| Tabla 8 CU_01..... | 27 |
| Tabla 9 CU_02..... | 27 |
| Tabla 10 CU_03..... | 27 |
| Tabla 11 CU_04..... | 28 |
| Tabla 12 CU_05..... | 28 |
| Tabla 13 CU_06..... | 28 |
| Tabla 14 CU_07..... | 29 |
| Tabla 15 CU_08..... | 29 |
| Tabla 16 CU_09..... | 30 |
| Tabla 17 CU_10..... | 30 |

LISTA DE ANEXOS

| | |
|--|----|
| ANEXO I Casos de uso del sistema de evaluación | 27 |
| ANEXO II Diagrama de clases del sistema de evaluación | 31 |
| ANEXO III Manual de usuario del sistema de evaluación | 32 |
| ANEXO IV Código fuente del sistema de evaluación | 32 |
| ANEXO IV Video del funcionamiento del sistema de valuación con la integración del primer componente..... | 32 |

RESUMEN

El presente componente desarrolló un sistema para la evaluación y visualización de los resultados de modelos de reconocimiento de 11 y 5 gestos de la mano (HGR). Es decir, la plataforma permite la carga de modelos HGR para su posterior evaluación y presentación de resultados tanto para clasificación y reconocimiento con gráficos dinámicos o dashboards que facilitan la comprensión de la evaluación. Este sistema de evaluación utilizó el modelo en cascada como parte del ciclo de desarrollo de software, respetando todas sus fases como fueron: definición de requerimientos, diseño, implementación, integración y pruebas del sistema y operación. Además, finalizado este proceso de desarrollo del producto software se realizó la integración con el primer componente a partir de un JSON de entrada con las predicciones del modelo HGR. El resultado final de este trabajo fue una aplicación web funcional e integrada con diferentes modelos HGR, el cual permite la evaluación de modelos de 5 y 11 gestos con diferentes números de usuarios y repeticiones respectivamente. Por último, hay que tomar en cuenta que se necesita configurar correctamente el archivo de entrada de los modelos HGR para su evaluación, caso contrario los resultados podrían no ser satisfactorios. Las evaluaciones de modelos HGR realizadas demuestran que se puede tener una alta precisión de clasificación, pero una baja precisión de reconocimiento por la manera que se evalúa para estos dos conceptos.

PALABRAS CLAVE: Cascada, gestos de mano, modelos HGR, clasificación, reconocimiento, dashboards.

ABSTRACT

This component developed a system for the evaluation and visualization of the results of recognition models of 11 and 5 hand gestures (HGR). In other words, the platform allows the loading of HGR models for their subsequent evaluation and presentation of results for both classification and recognition with dynamic graphics or dashboards that facilitate the understanding of the evaluation. This evaluation system used the cascade model as part of the software development cycle, respecting all its phases such as: definition of requirements, design, implementation, integration and testing of the system and operation. In addition, once this software product development process was completed, the integration with the first component was carried out from an input JSON with the predictions of the HGR model. The result of this work was a functional and integrated web application with different HGR models, which allows the evaluation of models of 5 and 11 gestures with different numbers of users and repetitions respectively. Finally, it must be considered that it is necessary to correctly configure the input file of the HGR models for its evaluation, otherwise the results may not be satisfactory. Likewise, the evaluations of HGR models carried out show that it is possible to have a high classification precision, however, it can have a low recognition precision due to the way the evaluation is carried out for these two different concepts.

KEYWORDS: Waterfall, hand gestures, HGR models, classification, recognition, dashboards.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En el presente componente se desarrolló un sistema para la evaluación y visualización de los resultados de modelos de reconocimiento de gestos HGR [1] . El sistema consiste en una aplicación web que permite la carga de archivos tipo JSON [2] de las predicciones de los modelos HGR. Posteriormente, el sistema evalúa la exactitud de clasificación y reconocimiento de los modelos HGR, cuyos resultados se visualizan en gráficos dinámicos o dashboards.

Como primer paso del desarrollo de este componente, se realizó la especificación del formato para la carga de los archivos JSON con sus respectivas predicciones. Luego, el sistema evaluó el modelo HGR y se obtuvo resultados para la exactitud de clasificación y reconocimiento [3]. La evaluación de clasificación consiste en obtener la etiqueta del gesto correctamente, en cambio la evaluación de reconocimiento consiste en determinar el momento exacto de realización del gesto. En consecuencia, pueden existir resultados con una correcta clasificación e incorrecto reconocimiento. Por último, para la interpretación de los resultados, se usaron diferentes gráficos para cada modelo evaluado.

Con el objetivo de construir esta aplicación web se utilizó el Modelo de Cascada con todas sus fases: definición de requerimientos, diseño del software y del sistema, implementación, integración y pruebas del sistema, operación. Finalmente, el sistema evaluó un modelo HGR que es el entregable del primer componente, para lo cual el modelo HGR debe seguir los lineamientos del formato JSON especificado.

1.1 Objetivo general

- Desarrollar un sistema para la evaluación y visualización de los resultados de modelos de reconocimiento de 11 gestos de la mano.

1.2 Objetivos específicos

- Establecer el formato de entrada JSON para la evaluación de modelos de reconocimiento de 11 gestos de la mano.
- Establecer los requisitos funcionales que tendrá el sistema de evaluación de modelos HGR.
- Desarrollar un sistema para la evaluación y visualización de los resultados de modelos HGR basado en el Modelo de Cascada.
- Evaluar el modelo HGR del primer componente mediante la aplicación web.

1.3 Alcance

En este componente se desarrolló el sistema de evaluación y visualización de las respuestas de modelos HGR. El modelo HGR debe reconocer 11 gestos de la mano, según se detalla en Tabla 1.

Tabla 1 Gestos de la mano

| Gesto | Gesto | Gesto |
|---|--|--|
| <p>Wave In</p>  | <p>Wave out</p>  | <p>Fist</p>  |
| <p>Open</p>  | <p>Up</p>  | <p>Pinch</p>  |
| <p>Down</p>  | <p>Left</p>  | <p>Right</p>  |
| <p>Backward</p>  | <p>Forward</p>  | |

La evaluación permitió conocer la exactitud tanto de la clasificación y reconocimiento de los gestos, de acuerdo con las predicciones de los modelos. Para lograr el desarrollo completo de la aplicación web se siguieron las fases del Modelo de Cascada expuestas a continuación:

- **Definición de requerimientos:**

En esta fase se obtuvo todos los requerimientos o casos de uso detallados del sistema, lo cual incluye: i) definición del formato JSON de entrada, ii) los requisitos para la evaluación de la exactitud de la clasificación y reconocimiento de las predicciones, iii) los requisitos de los dashboards y iv) restricciones del sistema.

- **Diseño del software y del sistema:**

Esta fase comprende el diseño de la arquitectura de software de la aplicación web en base a los requisitos establecidos en la fase previa.

- **Implementación:**

Esta fase empezó cuando el diseño de software del sistema concluyó dado que se utiliza una metodología en cascada. Esta fase consistió en el desarrollo de la aplicación web mediante la codificación de las diversas funcionalidades especificadas en los requerimientos.

- **Integración y pruebas del sistema:**

En esta fase se probó el sistema de evaluación integrado en su totalidad desde la carga de las predicciones JSON hasta la evaluación y visualización de resultados. Se realizaron pruebas de integración y aceptación. Se realizó todo este procedimiento para corroborar si se cumplen los requisitos especificados en la fase de requerimientos.

- **Operación:**

En esta última fase se buscaba cubrir los requisitos establecidos en la fase de definición de requerimientos y se evaluaron con el JSON de salida de predicciones del modelo del primer componente.

1.4 Marco teórico

1.4.1 Arquitecturas de software

La arquitectura de software se la define como la estructura de una aplicación o sistema, donde se relacionan componentes o elementos de software, además de la relación con las partes o propiedades externas que son visibles para el correcto funcionamiento de un producto de software [4].

Existen varios tipos de arquitectura, entre las más importantes se tienen las siguientes:

- Cliente-servidor
- Microservicios
- Por capas

Para este documento se detallará la arquitectura cliente-servidor elegida porque este patrón se suele escoger para aplicaciones web donde se tiene un cliente que realiza constantes peticiones al servidor.

1.4.2 Arquitectura cliente-servidor

Arquitectura de software en la que se tienen de dos partes explícitas como son el cliente y el servidor. El cliente realizará diferentes peticiones para obtener recursos para consumirlos y desplegarlos en la aplicación. Por otro lado, el servidor es el encargado de brindar los servicios o recursos para el correcto funcionamiento del programa [5].

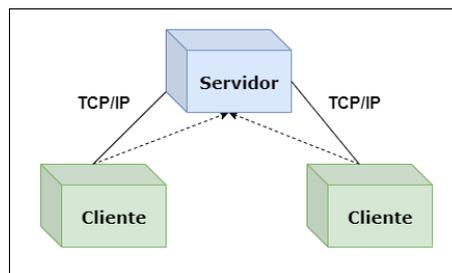


Figura 1 Arquitectura Cliente-Servidor

1.4.3 Bases de datos no relacionales

Las bases de datos no relacionales son modelos que almacenan estructuras de datos en diferentes formatos como pueden ser pares clave/valor, formato JSON o grafos, etc. [6]. Es decir, no se usa un modelo relacional o un esquema de tablas con relaciones entre estas. En su mayoría se usa para la carga masiva de datos en estructuras de datos más flexible para lograr el correcto funcionamiento de las aplicaciones.

1.4.4 Firebase

Plataforma de diversos servicios de Google para el desarrollo de aplicaciones de software. Está enfocada en brindar servicios a los desarrolladores para la construcción de productos de software de alta calidad. Este cuenta con servicios como: analítica, autenticación, bases de datos, nube, etc. [7].

- **Real time Database:** Base de datos no relacional en tiempo real que permiten la adaptación a aplicaciones de diversos lenguajes de programación. Permite la sincronización en tiempo real en el lado del cliente incluso cuando no existe conexión [8].

2 METODOLOGÍA

2.1 Selección metodología modelo en cascada

El modelo en cascada fue la metodología elegida para desarrollar la plataforma de evaluación, debido a que se tiene los requerimientos definidos desde el principio del plan de diseño de trabajo de integración curricular, además de no existir cambios en el desarrollo de software. Por otro lado, con dicho modelo se sigue una estructura secuencial con fases bien definidas y especificadas que se adaptan al contexto del componente realizado.

2.2 Definición JSON entrada

Para el comienzo del componente como se mencionó se definió la estructura del formato de entrada para el sistema de evaluación y el cual se detallará a continuación:

El archivo JSON contiene información necesaria para la evaluación de los resultados de los modelos de reconocimiento de gestos de la mano. Se tendrán dos formatos similares para modelos de 5 y 11 gestos. Este archivo debe seguir la estructura especificada en este documento, ya que caso contrario no se podrá realizar la evaluación. A continuación, se describen ambos formatos para el JSON de entrada.

Estructura 5 gestos:

Consideraciones:

- Las etiquetas validas son: ['waveIn', 'waveOut', 'fist', 'open', 'pinch', 'noGesture'].
- Esta estructura tiene la etiqueta “noGesture” como el gesto cuando está en relajamiento o sin hacer algún gesto.
- Número de usuarios: 306
- Número de repeticiones por usuario: 150

1. Primer nivel (Testing):

El archivo contiene un primer nivel denominado “testing”, haciendo referencia a que los resultados de los modelos se basaran en los datos de testeo y no de los de entrenamiento. Por lo que se debe predecir los usuarios de “testing”- “testing”.

2. Segundo nivel (Número de usuarios):

El segundo nivel tiene n usuarios de los que constan en “testing”, en el caso de los modelos de 5 gestos el número total será de 306 usuarios. Y el formato del nombre de dichos usuarios debe ser:

- user001...user306: Entonces de esta manera se incluyen los 306 usuarios testeados.

3. Tercer nivel (Clase, vector de etiquetas, vector de tiempos, vector de tiempos de procesamiento):

El tercer nivel contiene 4 claves:

- “class”: contiene una estructura anidada con el número de repeticiones (150) por usuario de la clase predicha.
 - La clave de la estructura anidada se representa por un índice y su valor que es la etiqueta predicha por el modelo:
Ej.: “idx_0”: “waveIn” – “idx_149”: “noGesture”
- “vectorOfLabels”: contiene a su vez otra estructura con 150 registros de cada repetición y un vector de etiquetas para determinar cuál es la etiqueta predicha.
 - Ej.: “idx_0”: [noGesture, noGesture, ..., pinch, pinch,... noGesture, noGesture]
- “vectorOfTimePoints”: de manera similar a la anterior contiene una estructura con 150 repeticiones representando el vector de etiquetas en tiempos.
 - Ej.: “idx_149”: [151,181,211,241, ...,811,841]
- “vectorOfProcessingTime”: De igual manera se tiene 150 repeticiones con un vector que representa los tiempos de procesamiento para cada etiqueta predicha en el vector de etiquetas.
 - Ej.: “idx_0”: [[0.1390356, ..., 0.0268871,0.0226709,0.0240634,0.0224788]

Estructura 11 gestos:

Consideraciones:

- Las etiquetas validas son: ['waveIn', 'waveOut', 'fist', 'open', 'pinch', 'up', 'down', 'left', 'right', 'forward', 'backward', 'relax']
- Esta estructura tiene la etiqueta “relax” como el gesto cuando está en relajamiento o sin hacer algún gesto.
- Número de usuarios: 42
- Número de repeticiones por usuario: 180

La estructura de 11 gestos es igual a la de 5 gestos solamente variando el nombre dado a los usuarios, número de usuarios y repeticiones. Además, de aumentar varios gestos.

- Los usuarios serán 42 y se denominarán: “user_001, ..., user_042”.
- El número de repeticiones serán: “idx_0, ..., idx_179”

- El gesto por defecto será: “relax”
- El tamaño del vector de etiquetas puede variar y deberán ser iguales al de tiempos, procesamiento.

2.3 Definición de requerimientos

Los requerimientos se fueron estableciendo en el transcurso de las reuniones del plan de diseño de trabajo de integración curricular. En donde se estableció los requisitos y las restricciones del sistema de la plataforma de evaluación.

2.3.1 Requerimientos funcionales

Los requerimientos funcionales fueron modelados siguiendo el formato de casos de uso y así poder plasmar las necesidades del usuario final. Los casos de uso siguen una estructura en la que consta su identificador, nombre, descripción, actores, detalle o pasos a realizar por parte del actor o software y por último excepciones que podrían existir.

Se modelaron diez casos de uso para representar todas las funcionalidades que posee el sistema de evaluación y de esta manera lograr validar la calidad del producto de software final. A continuación, se ejemplificará un caso de uso para entender el formato y los valores esperados en la Tabla 2:

Tabla 2 Caso de uso CU_01

| | | |
|----------------------------------|--|--|
| Identificador de Caso Uso | CU_01 | |
| Nombre | Ingresar a la plataforma de evaluación | |
| Descripción | Se podrá ingresar mediante una URL a la página web de la plataforma. | |
| Actores | Usuario | |
| Secuencia normal | | |
| Actor | Software | |
| 1. Ingresan la URL de la pagina | 2. Se presenta la página inicial del sistema. | |
| Excepciones | Software | |
| | | |

Los demás casos de uso se pueden visualizar en Anexo I.

2.3.2 Requerimientos no funcionales

El desarrollo de código fuente se realizó con Python y Typescript como lenguajes de programación. Además, se hizo uso de framework para el desarrollo más óptimo y rápido

como fueron Django y Nextjs respectivamente. Las versiones de ambos frameworks fueron las de 2023. Los IDE usados fueron Pycharm y WebStorm respectivamente.

2.4 Diseño

En esta fase se usaron los requerimientos para modelar la arquitectura necesitada en el sistema de evaluación. Además, se modeló el diagrama de clases requerido para construir el producto de software. Por último, se usó el diagrama de secuencias para entender correctamente el flujo que se iba a realizar en la aplicación web al momento de querer evaluar modelos, ver resultados, etc.

2.4.1 Diseño de arquitectura cliente-servidor

La arquitectura elegida para el sistema de evaluación fue la de cliente-servidor, ya que se dispone de un servidor centralizado y de varios clientes que iban a hacer peticiones. Además, esta arquitectura se tiene una evidente delegación de responsabilidades entre ambas partes. En la Figura 2 se observa cómo fue la arquitectura final.

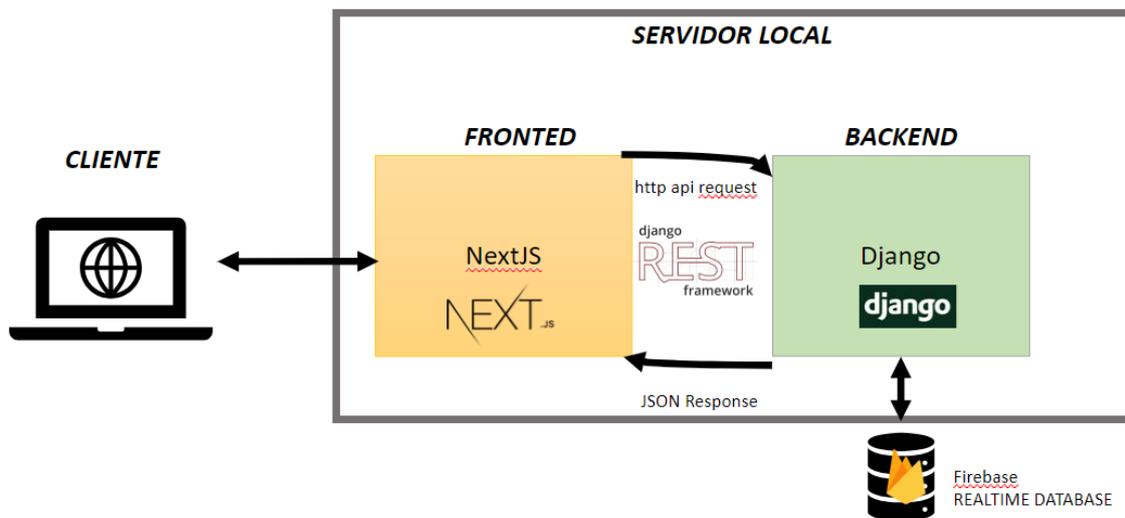


Figura 2 Arquitectura cliente-servidor del Sistema de Evaluación

Como se puede observar en la Figura 2 se implementó una arquitectura con un servidor local que tiene alojado el Backend (Django) y Frontend (React), en donde se comunican mediante un API-Rest de Django denominado “Django REST framework”. Así se pueden realizar las evaluaciones mediante peticiones del cliente al Frontend que a su vez delega estas responsabilidades al backend de las evaluaciones.

Por otro lado, todos los resultados de las evaluaciones se guardan en una base de datos en la nube de Firebase. La base de datos es una no relacional denominada “Real Time Database”, la cual nos permite tener actualizaciones de la información en tiempo real.

2.4.2 Diagrama de clases

El diagrama de clases se diseñó con el lenguaje UML para más legibilidad de las relaciones entre las diversas clases. Adicionalmente se usaron patrones de diseño “Factory” y “Singleton” para tener una clase en particular para la creación de los modelos y solo tener una instancia de evaluación para evitar procesos en paralelo respectivamente. El diagrama de clases se lo puede revisar con más detalle en el Anexo II. El diagrama de clases se compone de 5 clases y una clase abstracta.

- Clase Evaluator: Es la clase que se encarga de todo el proceso de evaluación de los diversos modelos, además de guardar características esenciales para la evaluación como el grupo de usuarios y repeticiones, el gesto por defecto, etc. Además, de todos los métodos para la evaluación por usuario, por repetición y la lógica para la representación de los resultados en gráficos dinámicos. En la Figura 3 se puede observar con más detalle.

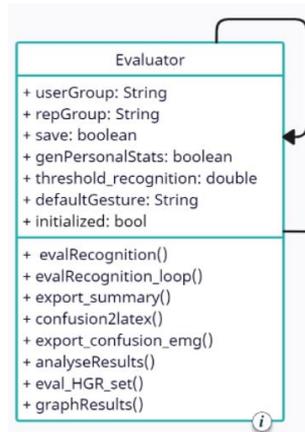


Figura 3 Clase Evaluator

- Clase Evaluated: Se encarga de recibir y enviar los modelos que se van a evaluar al evaluador y de esa manera tener una comunicación con dicho evaluador, en la Figura 4 siguiente se ven sus atributos y métodos.

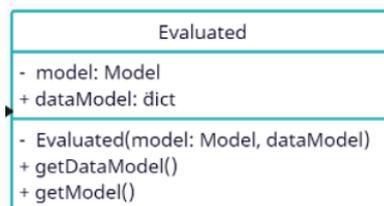


Figura 4 Clase Evaluated

Las siguientes clases al ser parte del patrón de diseño “Factory” se lo explicará como un todo y no por partes para mejor la comprensión del patrón. Para este patrón se

necesitó una clase abstracta modelo que representaría los modelos de entrada para el sistema de evaluación. Posteriormente, se hizo uso de dos clases “model11” y “model5” que heredan de la clase abstracta y son las clases de los modelos que se van a instanciar. Por último, el encargado de la creación de los objetos de cada clase fue el “ModelManager” facilitando de esta manera la escalabilidad de la creación de modelos si en algún momento se aumentara el número de gestos. En la Figura 5 se observa el patrón de diseño implementado.

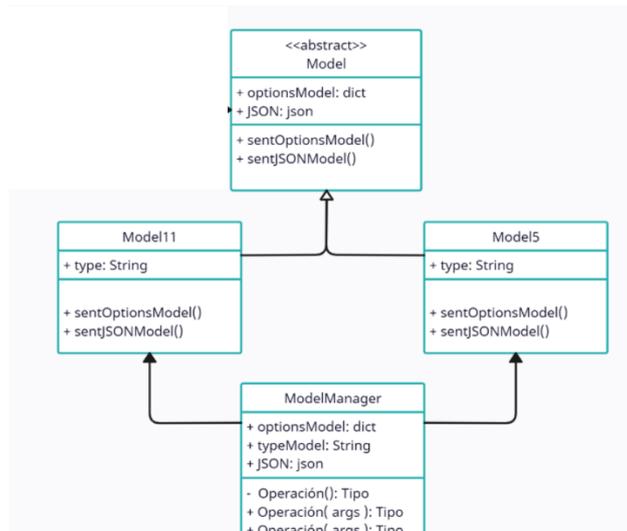


Figura 5 Patrón de diseño "Factory"

Por último, el patrón de diseño “Singleton” se usó en la clase evaluador como se observa en la Figura 3, ya que se necesitó solo tener una instancia de esta clase. Este patrón controla las evaluaciones en paralelo evitando resultados incorrectos de los modelos evaluados.

2.4.3 Diagrama de secuencias

Para un mejor entendimiento del flujo de actividades que se realizan en el sistema de evaluación se creó un diagrama de secuencias que empieza con un usuario evaluado enviando un formulario. Entonces el Frontend se encarga de procesar el JSON de entrada y delegar al Backend para que haga la evaluación del modelo y este mismo guarda los resultados de la evaluación en la base de datos. Luego cuando se requiera los resultados de las evaluaciones se recuperan de la base de datos. Al final el usuario evaluado puede visualizar los resultados de su evaluación. Este diagrama se presenta en la siguiente Figura 6.

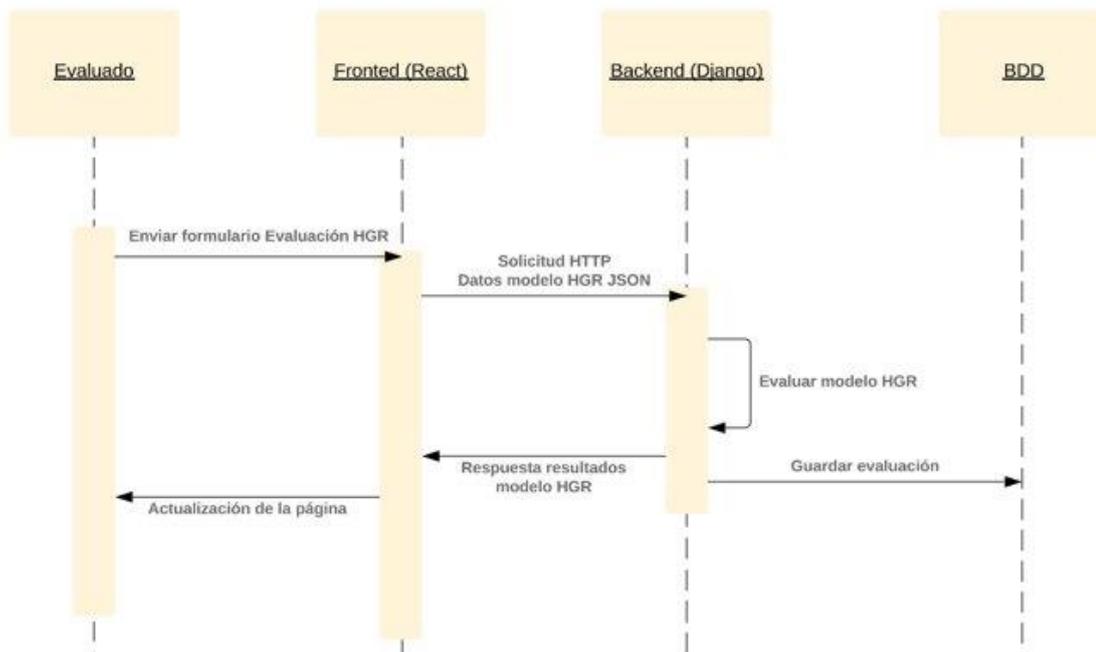


Figura 6 Diagrama de secuencias evaluación de un modelo

2.5 Implementación

Esta fase fue la más larga del proceso ya que se hizo toda la generación y desarrollo de código, tomando en cuenta los requerimientos y diseño definido en las fases anteriores. La codificación se la realizó en dos IDEs:

- WebStorm: IDE para el Frontend en su versión WebStorm 2023.1.1 en donde se hizo uso del framework Nextjs.
- Pycharm: IDE para el Backend en su versión Pycharm 2022.2.5 (Professional Edition) en donde se hizo uso del framework Django.

En cuanto al estándar de codificación se usó Camel Case, el nombre de atributos y métodos se mantuvo según el diseño y de esta manera poder garantizar la trazabilidad de todo el proceso del ciclo de desarrollo de software. Por otro lado, la aplicación web se implementó en el S.O Windows 11. Además, como se mencionó se usó dos framework que facilitaron el proceso de desarrollo:

- Django: Se uso la versión 4.1.7 para la parte del Backend y así mismo se usaron diferentes librerías, entre las cuales fueron Django Rest Framework, Numpy, Firebase-admin, sin embargo, la más importante que permitió la conexión entre el Backend y el Frontend fue Django Rest framework versión 3.14.0. El cual permitía

peticiones desde el Frontend para la consulta de modelos y resultados de evaluaciones.

- Nextjs: Se uso la versión 13.2.4 para la parte del Frontend, el cual es un framework para el trabajo con React. La ventaja de este es la facilidad de la renderización, compatibilidad con Typescript, etc.

Por otro lado, se hizo uso de GitHub para el versionamiento del código en donde ambas partes tanto el Frontend como el Backend se tenía el mismo repositorio¹. La versión final de la aplicación se mantiene en la rama principal. Sin embargo, se crearon diferentes ramas para cada funcionalidad completa que se tenía. Un claro ejemplo fue usar dos ramas para la evaluación de 11 y 5 gestos, ya que al finalizar se integraron ambas funcionalidades en la rama principal.

2.6 Integración y pruebas del sistema

Para corroborar el cumplimiento de los requerimientos tras desarrollar el sistema de evaluación, se realizaron pruebas enfocadas a la integración de módulos o funcionalidades que la aplicación final debía tener. Adicional otras pruebas de aceptación en donde si se tuvo que evidenciar el cumplimiento de los casos de uso establecidos.

2.6.1 Pruebas de integración

Las pruebas de integración se definieron en base a casos que facilitan la evidencia de la unión de las funcionalidades del sistema de evaluación. Estas pruebas están enfocadas a la unión del backend con el Frontend de la aplicación, la cual se realiza por medio de peticiones HTTP. Dichos casos de prueba cuentan con una descripción y un resultado esperado. Estos casos de prueba se muestran a continuación en la Tabla 3:

Tabla 3 Casos de prueba

| ID | Módulos o funcionalidades | Caso de Prueba | Descripción del caso de Prueba | Resultado esperado |
|----|---|---|--------------------------------|--|
| 1 | Petición obtener modelos evaluados desde el Frontend al backend | Verificar la información de los modelos evaluados en la pestaña "Rank Models" | Seleccionar "Rank models" | El sistema debe mostrar todos los modelos evaluados. |

¹Enlace al repositorio del proyecto: https://github.com/laboratorioAI/2023_HGR_Evaluation_APP

| | | | | |
|---|---|---|---|---|
| 2 | Petición obtener modelo evaluado desde el Frontend al backend | Verificar la información del modelo evaluado con todos sus resultados. | Seleccionar "Rank models" Seleccionar botón del modelo. | El sistema debe mostrar el modelo evaluado con sus resultados y gráficos dinámicos. |
| 3 | Petición evaluar modelo de 11 gestos desde el Frontend al backend | Verificar la evaluación del modelo de 11 gestos y los resultados obtenidos. | Seleccionar "Submit" Seleccionar modelo de 11 gestos y subir el JSON de entrada Enviar formulario | El sistema debe evaluar el modelo y presentar los resultados resumidos obtenidos. |
| 4 | Petición evaluar modelo de 5 gestos desde el Frontend al backend | Verificar la evaluación del modelo de 5 gestos y los resultados obtenidos. | Seleccionar "Submit" Seleccionar modelo de 5 gestos y subir el JSON de entrada Enviar formulario | El sistema debe evaluar el modelo y presentar los resultados resumidos obtenidos. |

2.6.2 Pruebas de aceptación

Para finalizar la fase de integración y pruebas del sistema se realizaron un conjunto de pruebas de aceptación que se basaron en los requerimientos especificados en una fase previa. Por tal razón, estas pruebas se basan en los casos de prueba detallados en el Anexo I. La Tabla 4 siguiente muestra estas pruebas.

Tabla 4 Pruebas de aceptación

| Id | Caso de uso | Resultado esperado |
|-----------|--|---|
| 1 | Ingresar a la plataforma de evaluación | El sistema de evaluación debe mostrar la pantalla de "Home" |
| 2 | Visualizar modelos evaluados | El sistema debe mostrar todos los modelos evaluados en la pestaña de "Rank models" |
| 3 | Visualizar modelo evaluado | El sistema debe mostrar los resultados de la evaluación y gráficos dinámicos de un modelo. |
| 4 | Descargar modelo evaluado en formato .csv. | El sistema debe entregar un archivo csv con los resultados del modelo. |
| 5 | Visualizar gráficos dinámicos de clasificación, reconocimiento y tiempos | El sistema debe mostrar gráficos dinámicos de cierto modelo. |
| 6 | Descargar gráfico de clasificación, reconocimiento o tiempo | El sistema debe entregar archivos png con los gráficos requeridos. |
| 7 | Visualizar formulario para la evaluación del modelo | El sistema debe mostrar la pantalla de "Submit" para la evaluación del modelo. |
| 8 | Evaluar modelo en la plataforma | El sistema debe obtener la información del formulario y posteriormente evaluar el modelo y presentar resultados resumidos en la pantalla. |
| 9 | Visualizar pantalla de ayuda | El sistema debe mostrar la pantalla "Help". |

| | | |
|----|--|--|
| 10 | Enviar correos para conocer el estado de la evaluación | El sistema debe enviar correos de notificaciones del estado de la evaluación del modelo. |
|----|--|--|

2.7 Operación

Una vez terminadas todas las pruebas del sistema se realizó el proceso de implementación de la aplicación en donde se siguieron ciertos pasos para su correcto funcionamiento:

- Primero se preparó y optimizó el código para su eficaz operación en producción, por tanto, se tuvo que probar en un ambiente de desarrollo la aplicación tanto la parte del backend y Frontend.
 - Backend: Hay que tomar en cuenta que para usar Python se configuró un entorno virtual o un interpretador de Python, el que tiene todos los paquetes necesarios y su versión para el uso del aplicativo. De igual manera el archivo “requirements.txt” tiene todas las dependencias necesarias para la parte del backend. Los pasos para levantar la aplicación en desarrollo fue la siguiente:
 - Dirigirse a la dirección del directorio del backend, el cual tiene el mismo nombre mencionado.
 - Ejecutar el comando “python .\manage.py runserver”

Luego se podrá observar cómo se levantó el servidor local para el ambiente de desarrollo como se muestra en la Figura 7.

```
(Backend) PS C:\Users\59399\Documents\TESIS\EvaluationSystemaHGR11\Backend> python .\manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 14 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes.
Run 'python manage.py migrate' to apply them.
August 05, 2023 - 12:41:55
Django version 4.1.7, using settings 'Backend.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figura 7 Levantamiento servidor local Backend

- Frontend: En este caso al usar Node.js todas las dependencias necesarias se encuentran en el directorio Frontend en un archivo denominado “package.json” y para obtener todas las librerías se debe seguir los siguientes pasos:
 - Ubicarse en el directorio del Frontend y acceder al directorio “.evaluation-system-hgr\”.

- Ejecutar el comando “npm install”, el cual instala todas las dependencias necesarias y crea el directorio node_modules.
- Ejecutar el comando “npm run dev” el mismo que permite levantar el ambiente de desarrollo del Frontend.

En la Figura 8 se muestra un claro ejemplo de este proceso para levantar el servidor.

```
PS C:\Users\59399\Documents\TESIS\EvaluationSystemaHGR11\Frontend> cd .\evaluation-system-hgr\
PS C:\Users\59399\Documents\TESIS\EvaluationSystemaHGR11\Frontend\evaluation-system-hgr> npm run dev

> evaluation-system-hgr@0.1.0 dev
> next dev

ready - started server on 0.0.0.0:3000, url: http://localhost:3000
warn - Your project has '@next/font' installed as a dependency, please use the built-in 'next/font' instead. The '@next/font' package will be removed in Next.js 14. You can migrate by
running 'npx @next/codemod@latest built-in-next-font'. Read more: https://nextjs.org/docs/messages/built-in-next-font
(node:3576) ExperimentalWarning: The Fetch API is an experimental feature. This feature could change at any time
(Use 'node --trace-warnings ...' to show where the warning was created)
event - compiled client and server successfully in 1616 ms (172 modules)
wait - compiling...
event - compiled successfully in 96 ms (149 modules)
```

Figura 8 Levantamiento servidor local Frontend

Una vez los dos servidores ya se encuentran levantados en el ambiente de desarrollo se ingresa la dirección URL que se muestra en la figura 8 para acceder al sistema de evaluación en un navegador. Un ejemplo de cómo se visualiza la aplicación web se muestra en la figura 9.

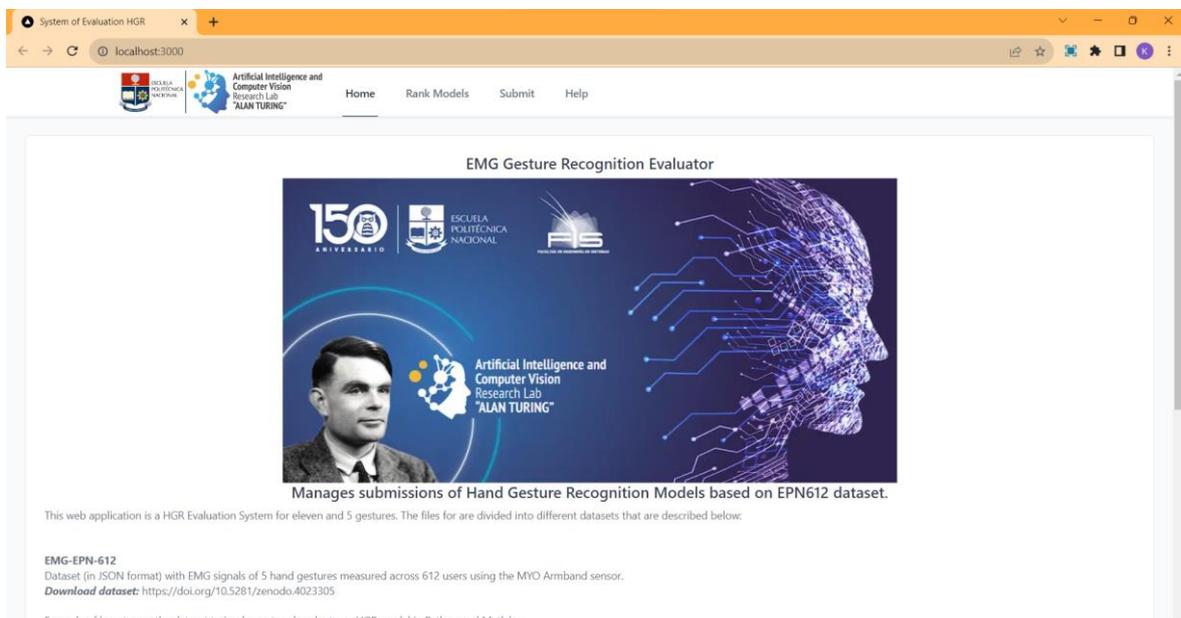


Figura 9 Pantalla principal sistema de evaluación HGR (Hand Gesture Recognition)

En caso de que el servidor backend no funcione o no se haya levantado correctamente y se ingrese a la dirección URL se podrá acceder a la página web, sin embargo, solo se muestra un mensaje de sin servicio actualmente. En la siguiente figura 10 se muestra el caso.

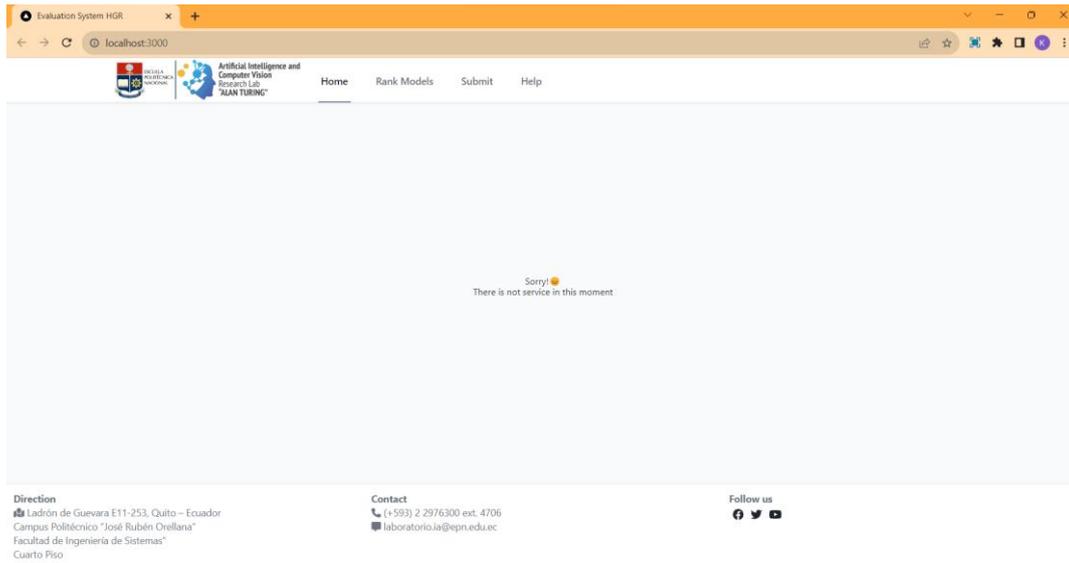


Figura 10 Página web sin servicio.

- Tras corroborar que el sistema de evaluación en el ambiente de desarrollo funciona correctamente, se pasará al despliegue para el ambiente de producción. Por tal razón, en la parte del Frontend se debía compilar el código y generar los archivos necesarios para esta nueva fase. Los pasos necesarios para lograrlo se detallan a continuación:
 - Ejecutar el comando “npm run build” el cual se encarga de compilar y crear un directorio “/.next” con los archivos optimizados para su renderización.
 - Ejecutar el comando “npm run start” para levantar la aplicación en el entorno de producción.

Los pasos realizados se muestran en la Figura 11 donde se hace este proceso en el servidor.

```
PS C:\Users\59399\Documents\TESIS\EvaluationSystemaHGR11\Frontend\evaluation-system-hgr> npm run build
> evaluation-system-hgr@0.1.0 build
> next build

info - Linting and checking validity of types
info - Compiled successfully
info - Collecting page data
├─┬ /analyze_results/[model] (1128 ms) 126 kB 234 kB
├─┬ /help (932 ms) 3.21 kB 107 kB
├─┬ /models (980 ms) 1.93 kB 125 kB
└─┬ /submit_model (1093 ms) 62 kB 180 kB
    * First Load JS shared by all 105 kB
      ├ chunks/framework-2c79e2a64abdb08b.js 45.2 kB
      ├ chunks/main-0ecb9ccfcb6c9b24.js 27 kB
      ├ chunks/pages/_app-21b523a66ac38962.js 307 B
      ├ chunks/webpack-59c5c889f52620d6.js 819 B
      └ css/ec59948588712160.css 31.5 kB

┌ (Static) automatically rendered as static HTML (uses no initial props)
└ info - Creating an optimized production build ..
PS C:\Users\59399\Documents\TESIS\EvaluationSystemaHGR11\Frontend\evaluation-system-hgr> npm run start
> evaluation-system-hgr@0.1.0 start
> next start

ready - started server on 0.0.0.0:3000, url: http://localhost:3000
```

Figura 11 Proceso de compilación y levantamiento fase de producción

El resultado esperado es el mismo que se obtuvo en la fase de desarrollo al momento de levantar los servidores tanto para el Backend y Frontend mostrado en la Figura 10. De esta manera se tiene un aplicativo funcional y que ha pasado por todo el ciclo de desarrollo de Software corroborando su calidad y durabilidad.

2.8 Evaluación con el modelo de reconocimiento de gestos

Al finalizar el componente sobre el desarrollo del sistema de evaluación de gestos HGR fue necesario integrar con el primer componente. La manera que se logró unirlos fue mediante la evaluación de un modelo de gestos realizado en el primer componente como se muestra en la Figura 12. El primer componente realizó un modelo de predicciones de gestos y obtuvo ciertos resultados guardados en un archivo JSON. El sistema permitió cargar el archivo para evaluar dichas predicciones para clasificación y reconocimiento, de donde se obtuvieron resultados con sus respectivos gráficos dinámicos que detallan mejor la evaluación. En el siguiente apartado de resultados se evidencia dicha integración.

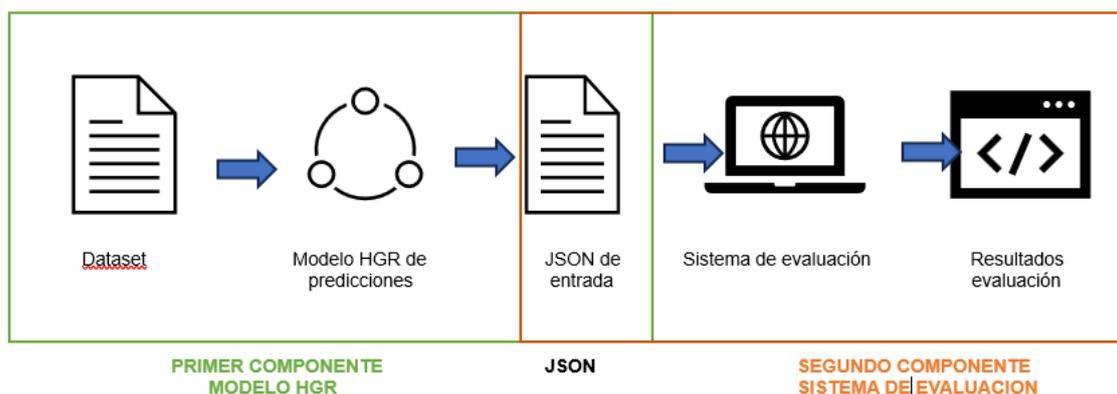


Figura 12 Integración primer y segundo componente

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

Los resultados del componente se basan en las pruebas realizadas desde la integración del sistema completo, pruebas de aceptación y por último la evaluación del primer componente. Las pruebas otorgaron evidencia de la calidad y eficiencia del producto de software realizado y por eso se resumen las pruebas con sus respectivos resultados.

En primer lugar, las pruebas de integración se basaron en la unificación del sistema de evaluación de sus diferentes módulos o funciones. Estas pruebas se especificaron en la

sección 2.6.1 y los resultados se pueden apreciar en la Tabla 5 en donde se muestra el identificador de la prueba de integración y el resultado obtenido.

Tabla 5 Resultados de las pruebas de integración

| ID | Resultado obtenido |
|----|--|
| 1 | El sistema de evaluación muestra un resumen de todos los modelos evaluados anteriormente con información correcta. |
| 2 | El sistema de evaluación muestra un modelo específico con todos los resultados de clasificación, reconocimiento y tiempos. También muestra gráficos dinámicos de todos los resultados. |
| 3 | El sistema carga y evalúa el modelo de 11 gestos y presenta de manera resumida los resultados obtenidos. |
| 4 | El sistema carga y evalúa el modelo de 5 gestos y presenta de manera resumida los resultados obtenidos. |

La tabla 5 muestra resultados muy satisfactorios, de acuerdo a los resultados especificados en las pruebas de integración. Por lo anterior mencionado se evidencia como se dio una eficaz integración y acoplamiento de los módulos y funcionalidades desarrolladas para el sistema de evaluación.

En segundo lugar, las pruebas de aceptación se basaron en los casos de uso especificados en la sección 2.3.1 y detallados en la sección de 2.6.2 donde se determinaron las pruebas de aceptación con el resultado esperado. Los resultados obtenidos de estas pruebas se presentan en la Tabla 6 mediante un identificador de la prueba de aceptación y el resultado esperado.

Tabla 6 Resultados obtenidos pruebas de aceptación

| ID | Resultado obtenido |
|----|--|
| 1 | El sistema de evaluación muestra la pantalla de "Home" |
| 2 | El sistema muestra todos los modelos evaluados en la pestaña de "Rank models" |
| 3 | El sistema muestra los resultados de la evaluación y gráficos dinámicos de un modelo. |
| 4 | El sistema entrega un archivo .csv con los resultados del modelo. |
| 5 | El sistema muestra gráficos dinámicos de cierto modelo. |
| 6 | El sistema entrega archivos .png con los gráficos requeridos. |
| 7 | El sistema muestra la pantalla de "Submit" para la evaluación del modelo. |
| 8 | El sistema obtiene la información del formulario y posteriormente evalúa el modelo y presenta resultados resumidos en la pantalla. |
| 9 | El sistema muestra la pantalla "Help". |
| 10 | El sistema envía correos de notificaciones del estado de la evaluación del modelo. |

La tabla 6 demuestra que los requerimientos y necesidades del cliente se han cumplido en su totalidad y es un software completo y funcional. Esto debido a que todas las pruebas de aceptación obtuvieron el mismo resultado que el esperado.

En tercer lugar, para finalizar el apartado de resultados se integraron ambos componentes y por tal motivo se evaluó mediante ciertas pruebas de aceptación que se especificaron en la sección 2.6.2. Las cuales tienen que ver con la carga y evaluación de modelos. En este caso la evaluación de un modelo de 11 gestos creado en el primer componente, los resultados obtenidos se desglosan en la Tabla 7.

Tabla 7 Resultados obtenidos pruebas de aceptación con el primer componente

| ID | Resultado obtenido |
|----|---|
| 7 | El sistema muestra la pantalla de "Submit" para la evaluación del modelo. |
| 8 | El sistema obtiene la información del formulario y posteriormente evalúa el modelo de 11 gestos del primer componente y presenta resultados resumidos en la pantalla. |
| 3 | El sistema muestra los resultados de la evaluación y gráficos dinámicos del modelo creado en el primer componente. |
| 4 | El sistema entrega un archivo .csv con los resultados del modelo del primer componente. |
| 5 | El sistema muestra gráficos dinámicos del modelo del primer componente. |
| 6 | El sistema entrega archivos .png con los gráficos requeridos. |

Para finalizar se muestran evidencias de esta integración desde la carga del modelo en la pantalla "Submit" en la Figura 12. El proceso de evaluación de los usuarios del modelo en la Figura 14 y el detalle resumido de los resultados obtenidos para el modelo evaluado en la Figura 15. Por último, los gráficos obtenidos de clasificación, reconocimiento y tiempos en las Figuras 16,17,18 respectivamente.

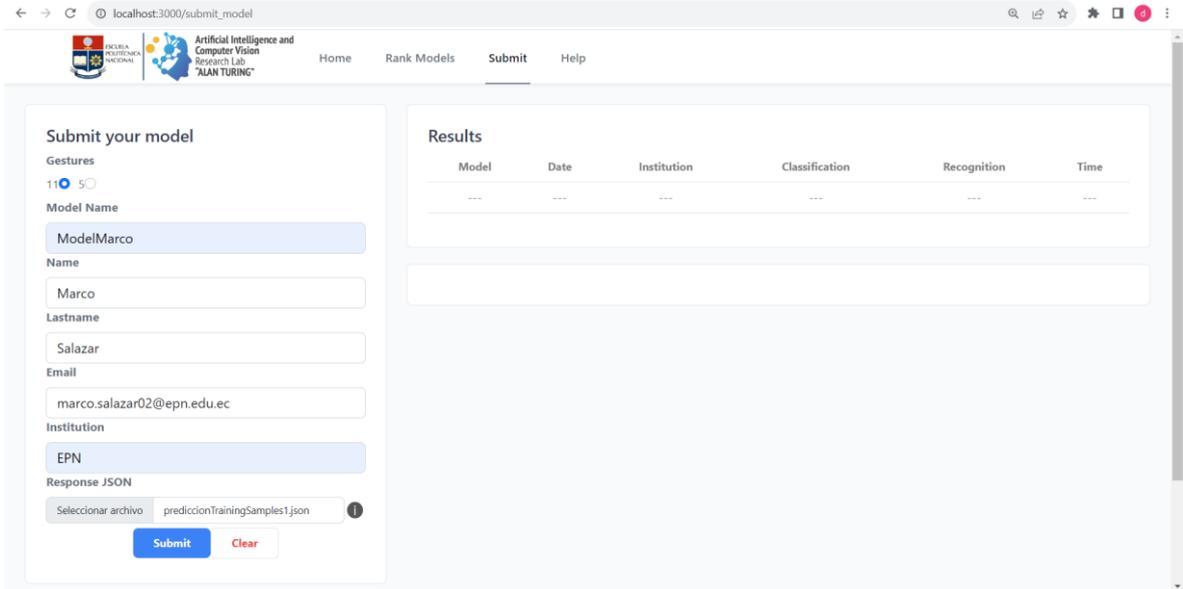


Figura 13 Pantalla Submit con modelo cargado

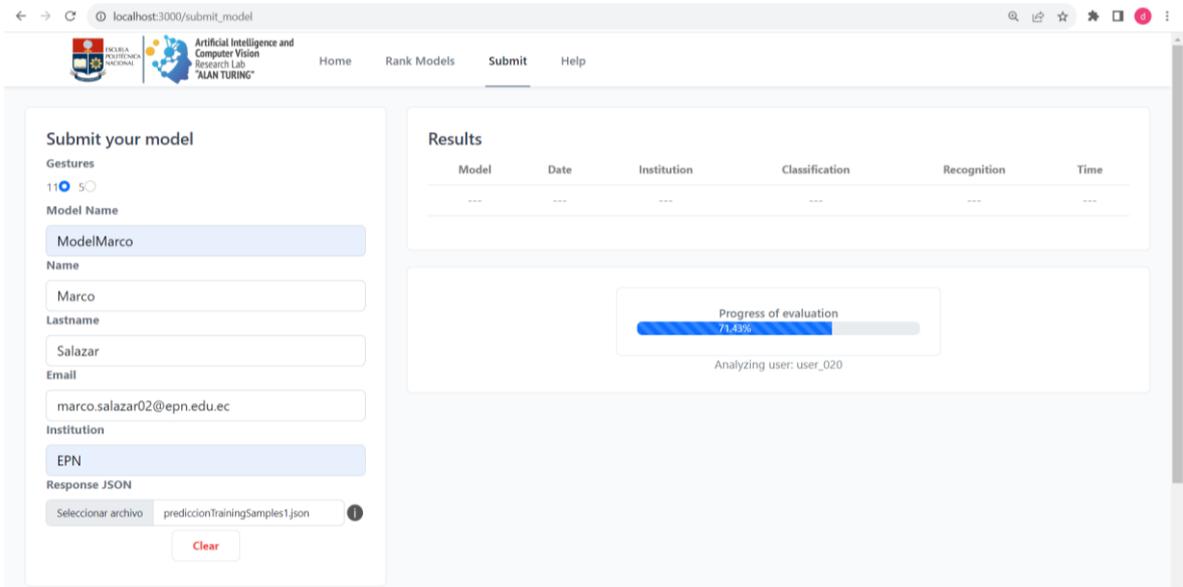


Figura 14 Proceso de evaluación del modelo.

Submit your model

Gestures
11 50

Model Name
ModelMarco

Name
Marco

Lastname
Salazar

Email
marco.salazar02@epn.edu.ec

Institution
EPN

Response JSON
Seleccionar archivo predicciónTrainingSamples1.json

Submit Clear

Results

| Model | Date | Institution | Classification | Recognition | Time |
|------------|------------------|-------------|----------------|-------------|----------|
| ModelMarco | 2023-08-12:18:42 | EPN | 60.61% | 18.93% | 0.0025ms |

View Details

Figura 15 Resultados resumidos de la evaluación del modelo

Model Name: ModelMarco

| Data | Classification | Recognition | Time |
|--|--|---|--|
| Institution : EPN Date : 2023-08-12:18:42 | Mean: 60.61% Deviation Standard: 11.53% | Mean: 18.93% Deviation Standard: 7.91% | Mean: 0.25ms Deviation Standard: 0.13ms |

Classification graphic

Dispersion Histogram Confusion

Accuracy Promedio: 60.61% Desviación superior: 72.13% Desviación inferior: 49.08%

Figura 16 Resultados de clasificación del modelo evaluado

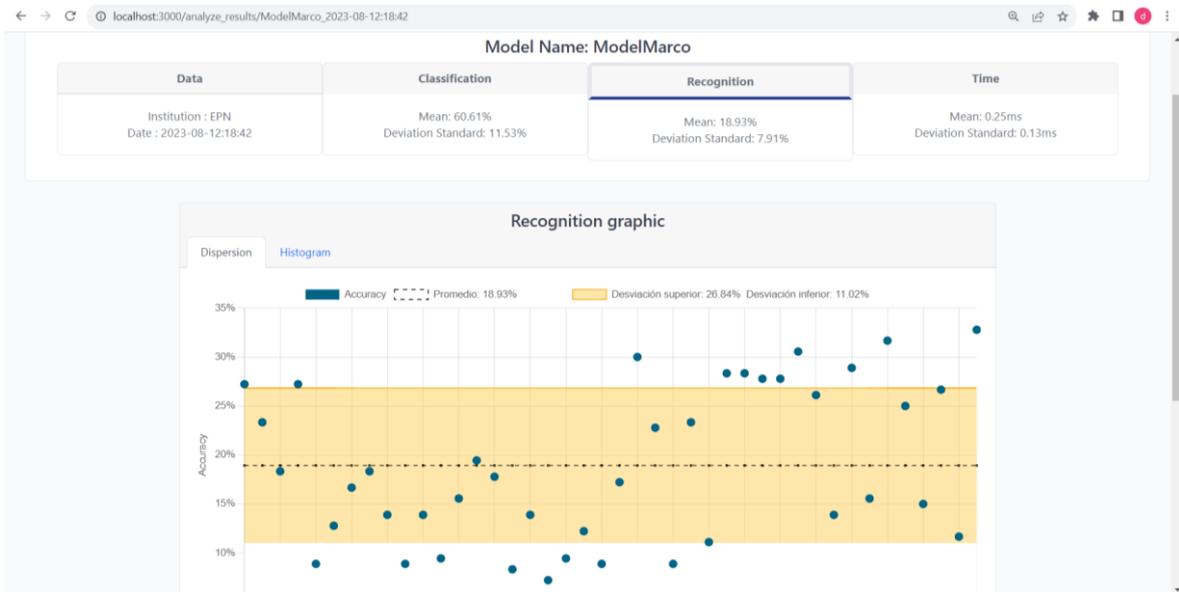


Figura 17 Resultados de reconocimiento del modelo evaluado

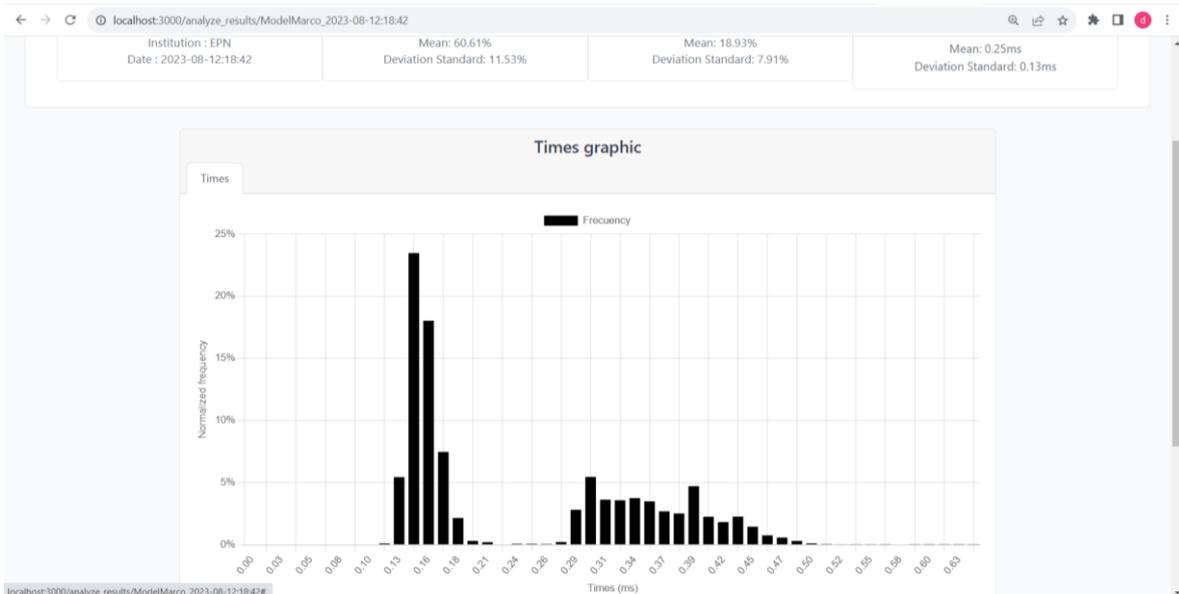


Figura 18 Resultados de tiempos del modelo evaluado

3.2 Conclusiones

El desarrollo del sistema de evaluación de modelos de reconocimiento de 11 gestos de la mano se diseñó e implementó correcta y satisfactoriamente. Esto se logró siguiendo el modelo de cascada con todas sus fases. Por tales razones, este proceso empezó con la definición de requerimientos completos para las siguientes fases, la misma que tuvo varios requisitos para avanzar a la fase de diseño del software. En primer lugar, se logró definir el JSON de entrada para el sistema de evaluación, luego los requerimientos funcionales, no funcionales y por último algunas restricciones que se establecieron en los requerimientos.

Además, las siguientes fases como el diseño, implementación, integración y pruebas y operación se evidencian en este documento de una manera secuencial y característico del modelo de cascada.

El sistema de evaluación de este componente se integró satisfactoriamente con un archivo JSON de entrada con los resultados de las predicciones para los 35 usuarios y sus respectivas repeticiones. Hay que tomar en cuenta que la evaluación de modelos HGR solo se puede llevar a cabo si los archivos JSON de entrada cumplen con el formato establecido en este componente. El formato al ser diferente podría dar resultados erróneos o no lograr la evaluación. Por dichas razones, fue fundamental establecer dicho formato y especificar en la misma plataforma de evaluación las restricciones del sistema. Por último, hay que aclarar que en los resultados de la evaluación se puede obtener una alta precisión para la clasificación, pero se puede tener una baja precisión para el reconocimiento. Esto se debe a que la clasificación solo se necesita saber la etiqueta predicha, sin embargo, para el reconocimiento se requiere saber el momento exacto de cuando se ha hecho y finalizado el gesto.

3.3 Recomendaciones

Se recomienda para plataformas de evaluación de modelos de reconocimiento de gestos usar un enfoque del modelo de cascada, siempre y cuando los requerimientos sean específicos y no cambiantes. Además, de ser un proceso de desarrollo secuencial respetando las fases del modelo, así se obtiene un producto software funcional que cumpla con los requisitos obtenidos por el cliente.

Para tener evaluaciones correctas de los modelos HGR se recomienda especificar detalladamente el formato de los archivos JSON de entrada, ya que una pequeña variación puede ocasionar resultados no precisos o hasta erróneos. Así mismo, los modelos entregados para su evaluación deben tener el número completo de usuarios y repeticiones, ya que generará un resultado veraz y correcto, sin sesgos por modelos sobrentrenados para ciertos usuarios o gestos en concreto.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] C. L. Huang, «A model-based hand gesture recognition system. Machine vision and applications,» 2001, pp. 243-258.
- [2] W. Jackson, «JSON quick syntax reference,» Apress., 2016.
- [3] S. Mitra, «Gesture recognition: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),» 2007, pp. 311-324.
- [4] L. Bass, Software architecture in practice, SEI Series in Software Engineering, Addison Wesley, 2003.
- [5] O. K. G. M. J. J. & G. A. Lizama, Redes de computadores Arquitectura Cliente-Servidor., Universidad Tecnica Federico Santa Maria, 2016.
- [6] «Microsoft Learn: Build skills that open doors in your career,» Microsoft, [En línea]. Available: <https://learn.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data>.
- [7] «Firebase,» Google, [En línea]. Available: <https://firebase.google.com/docs/database?hl=es-419>.
- [8] F. R. T. Database, «Firebase Realtime database,» Firebase, [En línea]. Available: <https://firebase.google.com/docs/database?hl=es-419>.

5 ANEXOS

| | |
|--|----|
| ANEXO I Casos de uso del sistema de evaluación | 27 |
| ANEXO II Diagrama de clases del sistema de evaluación | 31 |
| ANEXO III Manual de usuario del sistema de evaluación | 32 |
| ANEXO IV Código fuente del sistema de evaluación | 32 |
| ANEXO IV Video del funcionamiento del sistema de valuación con la integración del primer componente..... | 32 |

ANEXO I Casos de uso del sistema de evaluación

Tabla 8 CU_01

| | | |
|----------------------------------|--|---|
| Identificador de Caso Uso | CU_01 | |
| Nombre | Ingresar a la plataforma de evaluación | |
| Descripción | Se podrá ingresar mediante una URL a la página web de la plataforma. | |
| Actores | Usuario | |
| Secuencia normal | | |
| Actor | | Software |
| 1. Ingresan la URL de la pagina | | 2. Se presenta la página inicial del sistema. |
| Excepciones | | Software |
| | | |

Tabla 9 CU_02

| | | |
|----------------------------------|--|------------------------------------|
| Identificador de Caso Uso | CU_02 | |
| Nombre | Visualizar modelos evaluados | |
| Descripción | Se podrá ir a la sección de "Rank Models" mediante el banner y visualizar todos los modelos evaluados. | |
| Actores | Usuario | |
| Secuencia normal | | |
| Actor | | Software |
| 1. Selecciona "Rank Models" | | 2. Se presentan modelos evaluados. |
| Excepciones | | Software |
| | | |

Tabla 10 CU_03

| | | |
|--|--|--|
| Identificador de Caso Uso | CU_03 | |
| Nombre | Visualizar modelo evaluado | |
| Descripción | En la sección "Rank Models" se podrá visualizar información detallada de un modelo mediante la opción "Analyze". | |
| Actores | Usuario | |
| Secuencia normal | | |
| Actor | | Software |
| 1. Selecciona "Rank Models" | | 2. Se presentan modelos evaluados. |
| 3. Selecciona el icono de "analyze" de un modelo | | 4. Se presenta el modelo con información detallada y gráficos dinámicos. |
| Excepciones | | Software |
| | | |

Tabla 11 CU_04

| | | |
|---|--|--|
| Identificador de Caso Uso | CU_04 | |
| Nombre | Descargar modelo evaluado en formato csv. | |
| Descripción | En la sección "Rank Models" se podrá descargar información detallada de un modelo mediante la opción "Download". | |
| Actores | Usuario | |
| Secuencia normal | | |
| Actor | | Software |
| 1. Selecciona "Rank Models" | | 2. Se presentan modelos evaluados. |
| 3. Selecciona el icono de "download csv" de un modelo | | 4. Se descarga el modelo con información detallada en formato csv. |
| Excepciones | | Software |
| | | |

Tabla 12 CU_05

| | | |
|--|--|--|
| Identificador de Caso Uso | CU_05 | |
| Nombre | Visualizar gráficos dinámicos de clasificación, reconocimiento y tiempos | |
| Descripción | Se podrá ver gráficos dinámicos con información detallada de la evaluación de clasificación, reconocimiento y tiempos. | |
| Actores | Usuario | |
| Secuencia normal | | |
| Actor | | Software |
| 1. Selecciona "Rank Models" | | 2. Se presentan modelos evaluados. |
| 3. Selecciona el icono de "analyze" de un modelo | | 4. Se presenta el modelo con información detallada y gráficos dinámicos. |
| 5. Interactúa con el gráfico | | 6. Se muestra información dinámica del gráfico |
| 7. Selecciona "recognition" | | 8. Se actualiza información sobre la evaluación de reconocimiento. |
| 9. Selecciona "times" | | 10. Se actualiza información sobre los tiempos de evaluación. |
| Excepciones | | Software |
| | | |

Tabla 13 CU_06

| | | |
|----------------------------------|--|--|
| Identificador de Caso Uso | CU_06 | |
| Nombre | Descargar gráfico de clasificación, reconocimiento o tiempo | |
| Descripción | Se podrá descargar los gráficos de clasificación, reconocimiento o tiempos de tipo dispersión, histograma o confusión según corresponda. | |
| Actores | Usuario | |
| Secuencia normal | | |

| Actor | Software |
|--|--|
| 1. Selecciona "Rank Models" | 2. Se presentan modelos evaluados. |
| 3. Selecciona el icono de "analyze" de un modelo | 4. Se presenta el modelo con información detallada y gráficos dinámicos. |
| 5. Se desplaza al final del gráfico y se selecciona "Download Graphic" | 6. Se descarga el grafico en formato png dependiendo el grafico que se visualice en ese momento. |
| Excepciones | Software |
| Si se selecciona el grafico de tipo reconocimiento o de tiempos | No se puede descargar el grafico de la matriz de confusión. |

Tabla 14 CU_07

| Identificador de Caso Uso | CU_07 |
|----------------------------------|--|
| Nombre | Visualizar formulario para la evaluación del modelo |
| Descripción | Se podrá evaluar el modelo en el apartado de "Submit". |
| Actores | Usuario |
| Secuencia normal | |
| Actor | Software |
| 1. Selecciona "Submit" | 2. Se presenta el formulario para la evaluación. |
| Excepciones | Software |
| | |

Tabla 15 CU_08

| Identificador de Caso Uso | CU_08 |
|---|--|
| Nombre | Evaluar modelo en la plataforma |
| Descripción | Se podrá evaluar un modelo previamente llenando el formulario con todos los campos. |
| Actores | Usuario |
| Secuencia normal | |
| Actor | Software |
| 1. Selecciona "Submit" | 2. Se presenta el formulario para la evaluación. |
| 3. Se selecciona 11 o 5 gestos | |
| 4. Se llena los campos de nombre del modelo, nombre, apellido, email, institución | 5. Se generan mensajes de error si se llena el formulario incorrectamente. |
| 6. Se carga el archivo JSON de entrada con las predicciones del modelo | 7. Se genera mensajes de error si el JSON de entrada no cumple el formato esperado. |
| 8. Aplastar el botón de "Submit" | 9. Se empieza a cargar el archivo JSON en el sistema. |
| | 10. Se evalúa el modelo por cada usuario. |
| | 11. Se presenta una barra de carga con el tiempo faltante para terminar la evaluación. |

| | |
|---|---|
| | 12. Se muestra resultados de la evaluación. |
| Excepciones | Software |
| Se llena con datos erróneos el formulario | EL sistema presenta mensajes de error para controlar los errores. |

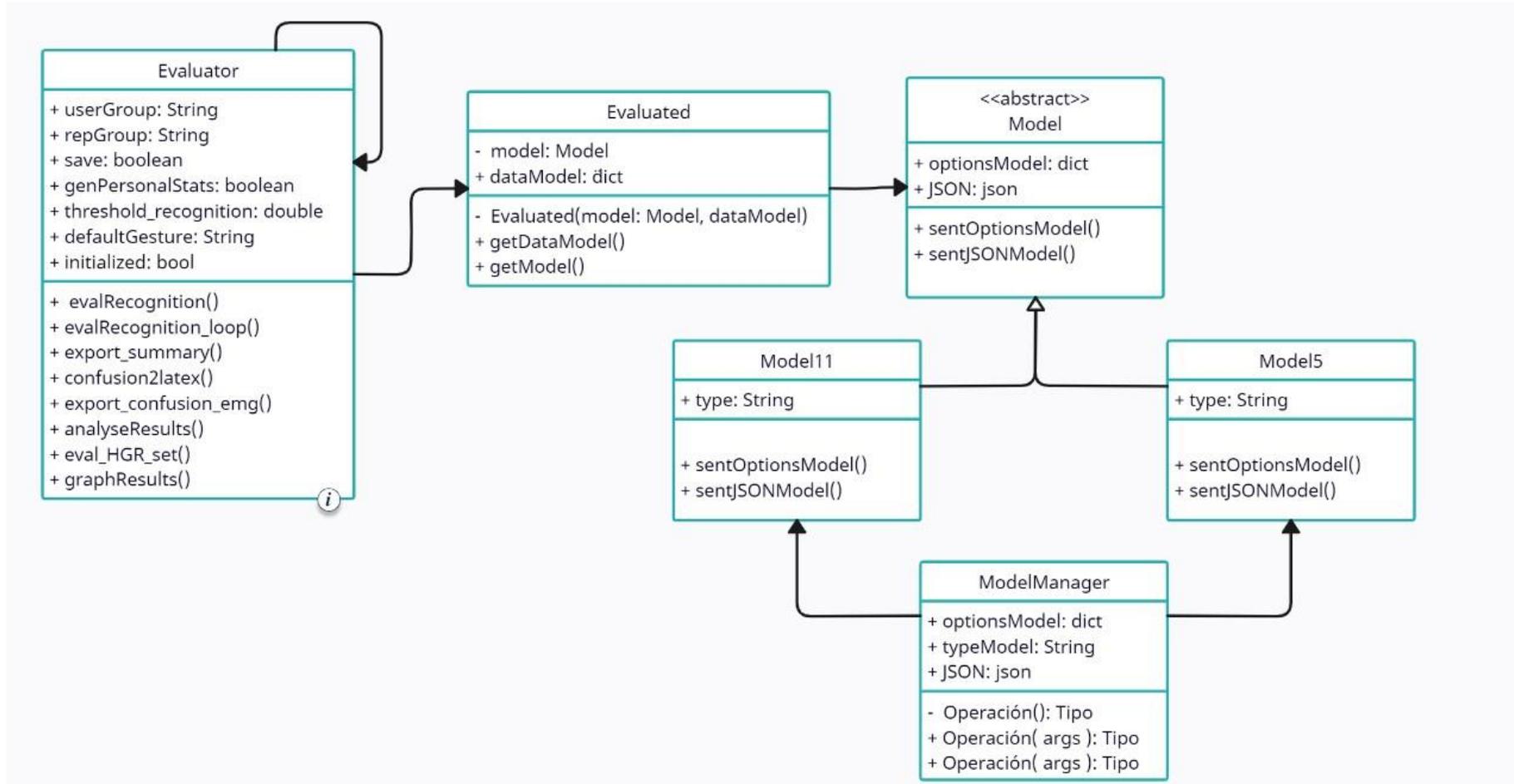
Tabla 16 CU_09

| | | |
|----------------------------------|---|-----------------------------------|
| Identificador de Caso Uso | CU_09 | |
| Nombre | Visualizar pantalla de ayuda | |
| Descripción | Se podrá ir la sección de "Help" y ver información de uso de la plataforma de evaluación. | |
| Actores | Usuario | |
| Secuencia normal | | |
| Actor | | Software |
| 1. Selecciona "Help" | | 2. Se presentan sección de ayuda. |
| Excepciones | | Software |
| | | |

Tabla 17 CU_10

| | | |
|---|--|--|
| Identificador de Caso Uso | CU_10 | |
| Nombre | Enviar correos para conocer el estado de la evaluación | |
| Descripción | Se podrá tener notificaciones por medio de correo para que el administrador sepa que modelos se están evaluando. | |
| Actores | Administrador, usuario | |
| Secuencia normal | | |
| Actor | | Software |
| 1. El usuario llena el formulario de evaluación. | | 2. El sistema envía un correo electrónico al administrador de la plataforma notificando el envío de un modelo. |
| | | 3. El sistema termina de evaluar y envía un correo electrónico con los resultados de la evaluación. |
| 4. El administrador revisa el correo electrónico con los resultados de la evaluación. | | |
| Excepciones | | Software |
| El usuario manda información errónea para evaluar en el archivo de entrada JSON | | El sistema envía un correo electrónico para notificar al administrador del evento. |

ANEXO II Diagrama de clases del sistema de evaluación



ANEXO III Manual de usuario del sistema de evaluación

[MANUAL DE USUARIO SISTEMA DE EVALUACION](#)

ANEXO IV Código fuente del sistema de evaluación

https://github.com/laboratorioAI/2023_HGR_Evaluation_APP

ANEXO IV Video del funcionamiento del sistema de valuación con la integración del primer componente

[VIDEO FUNCIONAMIENTO SISTEMA DE EVALUACIÓN](#)