AIWantTheJob

Plan de Pruebas

Jorge Alcalde Vesteiro

## Índice

[**Índice 2**](#_drvivphg8ptw)

[**Introducción 4**](#_numscet8oyf1)

[**Notas 4**](#_m2xrivw3so1l)

[**Enfoque de las pruebas 4**](#_t5x8zof1bcsx)

[**Sistema 5**](#_8uq9kqt8gahj)

[**Objetivos 5**](#_simmyimnndga)

[**Criterios de aceptación de las pruebas 5**](#_lthso0ghnqqe)

[**Criterios de suspensión 5**](#_wt4d328ooomv)

[**Organización 5**](#_6ht6fmfbk3i3)

[**Entorno 6**](#_42men7h9fwn3)

[**Entregables 6**](#_lhbtm6u2rpk3)

[**Pruebas Unitarias 6**](#_vz7hycpa98pl)

[Módulo: API 6](#_i2ds9exnnnhk)

[Resultados 8](#_a6zlj0fggjfi)

[Módulo: API Helper 8](#_o8xr4ooql5ww)

[Resultados 11](#_1rk1a9usrkfq)

[Módulo: Caché 11](#_owx7qkw90x7l)

[Resultados 12](#_p02acan6nf4w)

[Módulo: Database 12](#_72duno8ofz27)

[Resultados 14](#_h7htckk7uylo)

[Módulo: File Reader 14](#_1tpgndgjh7hj)

[Resultados 14](#_fhkgdhbt0i4y)

[Módulo: FTPclient 14](#_7supj7vkn526)

[Resultados 15](#_gfsu3dyamgmq)

[Módulo: FTPserverEu 15](#_lim1sujygcj)

[Resultados 17](#_2yuowhtm51oj)

[Módulo: LLM (Generic LLM) 17](#_15l0c2zc619)

[Resultados 18](#_fw0mnybeg1h)

[Módulo: main 18](#_gfoxi6n0lu9n)

[Resultados 18](#_xkbkb4j5jnsx)

[Módulo: Nemo Config 19](#_1vxr4yyhzrxu)

[Resultados 19](#_qkq2l1df603d)

[Módulo: Nemo Core 19](#_nsdkc7l9bdsb)

[Resultados 21](#_379iyr8pqaxg)

[Módulo: NumberGrabber 22](#_uhn2hgsqfk26)

[Resultados 22](#_fr12o2c4jrdh)

[Módulo: Parser 22](#_yoihs75dfuwz)

[Resultados 23](#_l033qdevy6ix)

[Módulo: PromptTemplates 23](#_m1pzcz90d2v0)

[Resultados 24](#_5pte2mapc1eg)

[Módulo: Proxy 24](#_5zol3eawryqm)

[Resultados 26](#_v817cdfse8kd)

[Módulo: VectoreStore 26](#_f3cbfch47cq3)

[Resultados 28](#_nixyq2yo0d94)

[Resultados Generales 28](#_5ova7kgs6i6i)

[**Pruebas de integración 28**](#_hnn2lqpnvm9h)

[**Pruebas de usabilidad 30**](#_l5dvn3duztue)

[Conclusiones del apartado 41](#_vhgmw5wn7ra8)

[**Pruebas end-to-end 42**](#_rsux0kc5rx4s)

[**Pruebas de despliegue 42**](#_vcpiw4im5d0c)

[Despliegue manual 42](#_xc7rhj1rlo04)

[Despliegue conda 42](#_1yh668ppet9z)

[**Conclusión 43**](#_av16g4jslei)

## Introducción

En este documento se presenta la documentación y la formalización de las pruebas realizadas para el Trabajo de fin de grado (TFG) de tipo B titulado “AIWantTheJob”.

En este documento se realizará una explicación de qué pruebas se han realizado y se formalizarán los resultados. El plan de pruebas abarca diferentes niveles de evaluación

1. Pruebas unitarias a nivel de componente
2. Pruebas de integración entre módulos
3. Pruebas de usabilidad del diseño del frontend
4. Pruebas de ejecución end to end
5. Pruebas de despliegue

Los resultados del testeos y de la aplicación se realizarán en base a los criterios definidos en el apartado de *criterios de aceptación de las pruebas*.

Características a Probar

Dada la naturaleza del proyecto (TFG) el alcance de la pruebas a realizar es amplio. Esto implica que se van a probar todos los componentes que forman parte del sistema a nivel unitario y a nivel de integración.

## Notas

1. Modularidad y dependencias: el nivel de modularidad del código es alto por lo que hay cierta dependencia entre módulos. Para solventar esa dependencia vamos a emplear Mocks que emulan el proceso.
2. Evaluación de IA: el empleo de técnicas de Inteligencia artificial implica que los resultados también se deberán evaluar en base a criterios semánticos. Eso significa que en las pruebas unitarias se emplea un modelo pequeño que devuelva texto cuando tenga que funcionar y en las pruebas de integración ya se empleará el modelo correcto puesto que en estas pruebas la evaluación semántica se hace manualmente.

## Enfoque de las pruebas

Se adoptará un enfoque exhaustivo explorando todos los aspectos del sistema diseñado, empezando a nivel de componente y llegando a nivel de sistema completo.

Cabe destacar que en las pruebas unitarias para este proyecto el acoplamiento entre caja blanca y caja negra es alto por lo que se empezará con las pruebas de caja negra y luego se realizará las pruebas de caja blanca de los procesos que no se hayan validado con la caja negra.

## Sistema

Para realizar las pruebas se va a emplear pytest para las unitarias, notebooks para las de integración y cuestionarios y un formulario de google con correcciones por otros usuarios para las de usabilidad.

Para ejecutar sobre todo las pruebas de integración se mantienen los requisitos mínimos de sistema mencionados en la memoria principal del sistema, 6 GB de VRAM.

## Objetivos

El objetivo principal de este plan de pruebas es verificar la robustez del código proporcionado y verificar que se cumple el criterio de aceptación A - 001.

Los resultados de las pruebas se recopilaron en este documento y se hará un overview de los mismos en la memoria a la que este plan va asociado.

Cualquier cambio que estas pruebas generen en el código será realizado según la gestión de configuración definida en la memoria

## Criterios de aceptación de las pruebas

Tal y como se menciona en la memoria y según el criterio de aceptación, las pruebas se pasarán sin se obtiene más del 90% de paso. Esto se aplica principalmente a las pruebas unitarias donde este porcentaje se puede contabilizar. Para el resto de pruebas, queda a disposición de los integrantes del proyecto decidir si la semántica y la aplicación en si funcionan según lo definido en los estándares

## Criterios de suspensión

En caso de que se detecte que más de un 30% de las pruebas fallan se habilita la opción a el encargado de las pruebas el poder parar la fase de testeo y volver a revisar el código puesto que dichas pruebas ya no se van a pasar.

## Organización

Estas pruebas están organizadas de la siguiente manera

1. Pruebas Unitarias
2. Pruebas de Integración
3. Pruebas de Usabilidad
4. Pruebas end-to-end
5. Pruebas de despliegue

## Entorno

Las pruebas se han realizado con las siguientes características

* Ubuntu 20.04
* Framework de PyTest (7.2.0 o superiores)
* Librerías de Jupyter Notebook

Puede ser que las pruebas necesiten acceso a la red si no se tiene descargados previamente los modelos a emplear sobre todo para las pruebas de integración.

## Entregables

Asociadas a la memoria del TFG AIWantTheJob está este pan de pruebas para simbolizar la realización de las pruebas. Dentro del repositorio del proyecto se encuentra un apartado de Tests donde se pueden verificar todas las pruebas.

## Pruebas Unitarias

Como ya se ha mencionado anteriormente, las pruebas unitarias sirven para ver la robustez a nivel de componente. Para realizar estas pruebas vamos a dividir los apartados por módulo, y dentro de cada módulo especificar las pruebas de Caja Negra y las de Caja Blanca. Esto se debe a que para nuestro sistema modular, las pruebas de caja negra cubren en su mayoría las pruebas de caja blanca.

En este caso, la mayoría de funciones tienen pocas opciones a probar y los caminos dependen de las opciones así que se validan juntas. Para ver las pruebas y ejecutarlas referir al apartado Tests/unit del repositorio anexado

### Módulo: API

| Función | get\_health | |
| --- | --- | --- |
| Descripción | Verificar el estado de salud de la API | |
| Inputs | Expected Output | Test ID |
| Válido: N/A | -Código de estado HTTP 200  -JSON response con campo 'status' que contiene 'available' o 'loading' | test\_get\_health\_status |
| No Válido: N/A | N/A | N/A |

| Función | subirarchivo | |
| --- | --- | --- |
| Descripción | Subir un archivo al servidor FTP y obtener datos. | |
| Inputs | Expected Output | Test ID |
| Válido: N/A | -Código de estado HTTP 200  -JSON response con los datos retornados por el servidor FTP | test\_subirarchivo |
| No Válido: N/A | N/A | N/A |

| Función | peticion | |
| --- | --- | --- |
| Descripción | Hacer una consulta al bot | |
| Inputs | Expected Output | Test ID |
| Válido: (args de la request) | -Código de estado HTTP 200  -JSON response con campos 'respuesta' y 'array' | test\_peticion |
| No Válido: N/A | N/A | N/A |

| Función | login | |
| --- | --- | --- |
| Descripción | Autenticar a un usuario | |
| Inputs | Expected Output | Test ID |
| Válido: user-password correctos | -Código de estado HTTP 200  -JSON response con campo 'id' conteniendo el ID del usuario | test\_login |
| No Válido: user-password incorrectos | -Código de estado HTTP 200  -JSON response con campo 'id' conteniendo None | test\_login\_fallo |

| Función | register | |
| --- | --- | --- |
| Descripción | Registrar un nuevo usuario | |
| Inputs | Expected Output | Test ID |
| Válido: user-password-password correctos | -Código de estado HTTP 200  -JSON response con campo 'id' conteniendo el ID del usuario | test\_register |
| No Válido: user ya existe | -Código de estado HTTP 200  -JSON response con campo 'id' conteniendo None | test\_register\_fallo |

| Función | array\_post | |
| --- | --- | --- |
| Descripción | Insertar un valor en el historial de puntuaciones de un usuario | |
| Inputs | Expected Output | Test ID |
| Válido: (argumentos de petición) | -Código de estado HTTP 200  -JSON response con campo 'array' conteniendo el array actualizado | test\_array\_post |
| No Válido: N/A | N/A | N/A |

| Función | array\_get | |
| --- | --- | --- |
| Descripción | Obtener el historial de valores de un usuario | |
| Inputs | Expected Output | Test ID |
| Válido: (argumentos de petición) | -Código de estado HTTP 200  -JSON response con campo 'array' conteniendo el array actualizado | test\_array\_get |
| No Válido: N/A | N/A | N/A |

| Función | \_\_init\_\_ | |
| --- | --- | --- |
| Descripción | Iniciar el API Server con sus valores de inicio | |
| Inputs | Expected Output | Test ID |
| Válido: valores de inicio | -Flask instance | test\_init |
| No Válido: N/A | N/A | N/A |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Módulo: API Helper

| Función | \_\_init\_\_ | |
| --- | --- | --- |
| Descripción | Iniciar el API helper con sus valores de inicio | |
| Inputs | Expected Output | Test ID |
| Válido: valores de inicio | -Tener las variables internas:  id = 1  \_ip = "127.0.0.1"  \_port = "5000"  API\_URL = "http://127.0.0.1:5000/peticion/"  HEALTH\_URL = "http://127.0.0.1:5000/health”  RAG\_URL = "http://127.0.0.1:5000/subirarchivo" | test\_init |
| No Válido: N/A | N/A | N/A |

| Función | set\_id | |
| --- | --- | --- |
| Descripción | Establecer el ID del usuario en el API Client | |
| Inputs | Expected Output | Test ID |
| Válido: “id\_nuevo” | -Tener las variables internas:  id = “id nuevo” | test\_set\_id |
| No Válido: N/A | N/A | N/A |

| Función | api\_query | |
| --- | --- | --- |
| Descripción | Realizar una consulta a la API | |
| Inputs | Expected Output | Test ID |
| Válido: diccionario payload y url | - Diccionario Resultado (Mockeado a {test:data} por ejemplo | test\_api\_query |
| No Válido: N/A  (Esta hardcodeado) | N/A | N/A |

| Función | query\_question | |
| --- | --- | --- |
| Descripción | Realizar una consulta de tipo pregunta a la API | |
| Inputs | Expected Output | Test ID |
| Válido: diccionario payload | - String Resultado (Mockeado a “test answer” por ejemplo | test\_query\_question |
| No Válido: N/A  (Esta hardcodeado) | N/A | N/A |

| Función | query\_example\_response | |
| --- | --- | --- |
| Descripción | Realizar una consulta de tipo ejemplo a la API | |
| Inputs | Expected Output | Test ID |
| Válido: diccionario payload | - String Resultado (Mockeado a “test answer” por ejemplo | test\_query\_example |
| No Válido: N/A  (Esta hardcodeado) | N/A | N/A |

| Función | query\_for\_grading | |
| --- | --- | --- |
| Descripción | Realizar una consulta de tipo evaluación a la AP | |
| Inputs | Expected Output | Test ID |
| Válido: diccionario payload | - String Resultado (Mockeado a “test answer” por ejemplo  - Array de valores (Mockeado) | test\_query\_evaluation |
| No Válido: N/A  (Esta hardcodeado) | N/A | N/A |

| Función | rag\_query | |
| --- | --- | --- |
| Descripción | Pedir los credenciales para poder subir los documentos a la API | |
| Inputs | Expected Output | Test ID |
| Válido: {Vacio} | - Retorna una tupla con los valores de 'address', 'port', 'user', y 'password' de la respuesta de la API | test\_rag |
| No Válido: N/A  (Esta hardcodeado) | N/A | N/A |

| Función | login | |
| --- | --- | --- |
| Descripción | Realizar una validación de usuario con el login | |
| Inputs | Expected Output | Test ID |
| Válido: credenciales correctos | - Retorna el valor ID | test\_login\_bien |
| No Válido: credenciales incorrectos | - Devuelve None | test\_login\_fallo |

| Función | register | |
| --- | --- | --- |
| Descripción | Realizar un registro de usuario con el register | |
| Inputs | Expected Output | Test ID |
| Válido: credenciales correctos | - Retorna el valor ID | test\_register\_bien |
| No Válido: credenciales incorrectos | - Devuelve None | test\_register\_fallo |

| Función | get\_array | |
| --- | --- | --- |
| Descripción | Realizar una petición para recuperar el array | |
| Inputs | Expected Output | Test ID |
| Válido: id existente | - Retorna el array de puntuaciones del usuario | test\_get\_array\_bien |
| No Válido: id no existe | - Devuelve [0, (0,0),] | test\_get\_array\_mal |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Módulo: Caché

| Función | get\_content | |
| --- | --- | --- |
| Descripción | Obtener el contenido cacheado y limpiar el caché | |
| Inputs | Expected Output | Test ID |
| Válido: se ha cacheado o se está cacheando la respuesta | - Retorna el diccionario respuesta | test\_get\_content |
| No Válido: no hay respuesta cacheada | - Devuelve None | test\_get\_content\_none |

| Función | prepare example | |
| --- | --- | --- |
| Descripción | Obtener el contenido cacheado y limpiar el caché | |
| Inputs | Expected Output | Test ID |
| Válido: pregunta | - Retorna el diccionario respuesta | test\_prepare\_example |
| No Válido: N/A está hardcodeado | N/A | N/A |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Módulo: Database

| Función | \_\_init\_\_ intrínseco con un fixture | |
| --- | --- | --- |
| Descripción | Iniciar la BD | |
| Inputs | Expected Output | Test ID |
| Válido: {argumentos por defecto} | - Se inicia | fixture |
| No Válido: N/A está hardcodeado | N/A | N/A |

| Función | \_generate\_identifier | |
| --- | --- | --- |
| Descripción | Generar un identificador único para un usuario | |
| Inputs | Expected Output | Test ID |
| Válido: {credenciales usuario y contraseña} | - un hash que empieza y termina por “ | test\_generate\_identifier |
| No Válido: N/A está hardcodeado | N/A | N/A |

| Función | \_hash\_password | |
| --- | --- | --- |
| Descripción | Generar el hash de la contraseña | |
| Inputs | Expected Output | Test ID |
| Válido: {contraseña} | - hash de la contraseña y de la sal | test\_\_hash\_password |
| No Válido: N/A está hardcodeado | N/A | N/A |

| Función | registrar\_usuario | |
| --- | --- | --- |
| Descripción | Registra el usuario en la BD | |
| Inputs | Expected Output | Test ID |
| Válido: {usuario y contraseña} | - ID del usuario | test\_registrar\_usuario\_success |
| No Válido: {usuario y contraseña existentes} | - string: “Name probably existing or internal error” | test\_registrar\_usuario\_existing\_user |
| No Válido: {Error} | - string: “Name probably existing or internal error” | test\_registrar\_usuario\_error |

| Función | validar\_usuario | |
| --- | --- | --- |
| Descripción | Realiza el login de usuario validando las credenciales en la BD | |
| Inputs | Expected Output | Test ID |
| Válido: {usuario y contraseña} | - ID del usuario | test\_validar\_usuario\_success |
| No Válido: {usuario y contraseña existentes} | - string: "Error on validation" | test\_validar\_usuario\_not\_found |
| No Válido: {Error} | - string: "Error on validation" | test\_validar\_usuario\_fail |

| Función | insertar\_valor\_array | |
| --- | --- | --- |
| Descripción | inserta un valor nuevo en el historial | |
| Inputs | Expected Output | Test ID |
| Válido: {id correcto, valor existente} | - Array de puntuaciones del usuario | test\_insertar\_valor\_array\_success |
| No Válido: {id incorrecto o valor inexistente} | - [(0, 0.0)] | test\_insertar\_valor\_array\_error |

| Función | recuperar\_valores\_array | |
| --- | --- | --- |
| Descripción | inserta un valor nuevo en el historial | |
| Inputs | Expected Output | Test ID |
| Válido: {id correcto} | - Array de puntuaciones del usuario | test\_recuperar\_valores\_array\_success |
| No Válido: {id incorrecto} | - [(0, 0.0)] | test\_recuperar\_valores\_array\_error |

| Función | close | |
| --- | --- | --- |
| Descripción | inserta un valor nuevo en el historial | |
| Inputs | Expected Output | Test ID |
| Válido: {Vacio} | - Se cierra la BD | test\_close |
| No Válido: N/A | N/A | N/A |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:

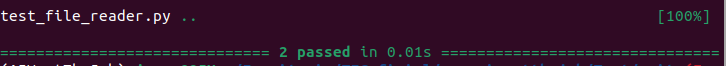


### Módulo: File Reader

| Función | read | |
| --- | --- | --- |
| Descripción | Convertir los bytes de un fichero en BytesIO | |
| Inputs | Expected Output | Test ID |
| Válido: bytes de un fichero | - Retorna el objeto BytesIO | test\_read |
| No Válido: no bytes o bytes incorrectos | - Devuelve un IOError | test\_read\_error |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Módulo: FTPclient

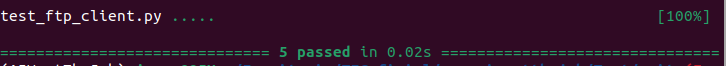
| Función | \_\_init\_\_ | |
| --- | --- | --- |
| Descripción | Inicializar el cliente FTP con los parámetros correctos | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se inician los parámetros  - address: "localhost"  - port: "9999"  - user: "user"  - password: "password" | test\_init |
| No Válido: {Error interno} | Error de FTPlib | test\_init\_connection\_error |

| Función | start | |
| --- | --- | --- |
| Descripción | Inicializar el cliente FTP (a runnear) | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se realizan los dos métodos internos | test\_start |
| No Válido: N/A | N/A | N/A |

| Función | upload\_file | |
| --- | --- | --- |
| Descripción | Subir un fichero | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se sube el fichero correcto (se llaman a las funciones indicadas) | test\_upload\_file |
| No Válido: {Error interno} | Error de FTPlib | test\_upload\_file\_error |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Módulo: FTPserverEu

| Función | \_\_init\_\_ | |
| --- | --- | --- |
| Descripción | Inicializar el servidor FTP con los parámetros correctos | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se inician los parámetros  - address: "localhost"  - port: "9999"  - user: "user"  - password: "password" | test\_init |
| No Válido: N/A | N/A | N/A |

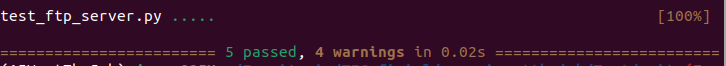
| Función | get\_data\_as\_dic | |
| --- | --- | --- |
| Descripción | Inicializar el cliente FTP con los parámetros correctos | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Devuelve un diccionario:  - address: "localhost"  - port: "9999"  - user: "user"  - password: "password" | test\_get\_data\_as\_dic |
| No Válido: N/A | N/A | N/A |

| Función | load | |
| --- | --- | --- |
| Descripción | Cargar el servidor con un target | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se llama a la función indicada | test\_load |
| No Válido: {Error interno} | FTPServer Error | test\_load\_error |

| Función | start | |
| --- | --- | --- |
| Descripción | Inicializar el servidor FTP (a runnear) | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se crea el hilo | test\_start |
| No Válido: N/A | N/A | N/A |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Módulo: LLM (Generic LLM)

| Función | \_\_init\_\_ | |
| --- | --- | --- |
| Descripción | Inicializar el LLM con los parámetros correctos | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se llama a la función de creación (AutoModelFromPretrained) | test\_llm\_initialization |
| Válido: {cuantizacion 4bit manual} | Se llama a la función de creación (AutoModelFromPretrained) | test\_llm\_start\_4bit |
| Válido: {cuantizacion 8bit manual} | Se llama a la función de creación (AutoModelFromPretrained) | test\_llm\_start\_8bit |
| Válido: {No cuentizacion manual} | Se llama a la función de creación (AutoModelFromPretrained) | test\_llm\_start\_no\_cuant |
| No Válido: {Error} | Error | N/A |

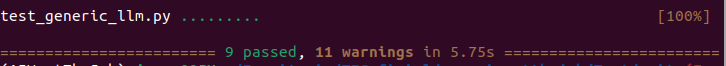
| Función | \_set\_default\_callbacks | |
| --- | --- | --- |
| Descripción | Configurar los callbacks por defecto | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se asignan los 3 callbacks y son Callables | test\_llm\_set\_default\_callbacks |
| No Válido: {N/A} | N/A | N/A |

| Función | \_set\_callback | |
| --- | --- | --- |
| Descripción | Se crea un callback | |
| Inputs | Expected Output | Test ID |
| Válido: {un sysprompt} | Se asignan el callback | test\_llm\_set\_callback |
| No Válido: {N/A} | N/A | N/A |

| Función | \_get\_chat\_template | |
| --- | --- | --- |
| Descripción | Se inicializa el chat template | |
| Inputs | Expected Output | Test ID |
| Válido: {un formato messages correcto} | Se asignan el chat template encontrado:  Mockeado: "Applied chat template" | test\_get\_chat\_template\_with\_chat\_template |
| Válido: {No hay chat template} | Autocorrección: Se aplica un chat template por defecto: Mockeado: "Applied chat template" | test\_get\_chat\_template\_without\_chat\_template |
| Válido: {Error} | Autocorrección: Se aplica un chat template por defecto: Mockeado: "Applied chat template" | test\_get\_chat\_template\_exception\_handling |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:

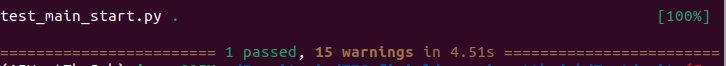


### Módulo: main

| Función | main | |
| --- | --- | --- |
| Descripción | Inicializa todo el backend | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se llaman a todas las funciones internas | test\_main\_script |
| No Válido: N/A | N/A | N/A |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



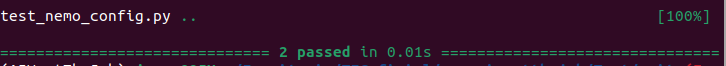
### Módulo: Nemo Config

| Función | \_\_init\_\_ | |
| --- | --- | --- |
| Descripción | Inicializar e NemoConfig (valida los 2 metodos internos) | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Se realizan los dos métodos internos | test\_init |
| No Válido: N/A | N/A | N/A |

| Función | \_set\_colang\_config (error) | |
| --- | --- | --- |
| Descripción | Instanciar la configuración Colang | |
| Inputs | Expected Output | Test ID |
| Válido: validado test\_init | N/A | N/A |
| No Válido: {fichero\_incorrecto} | FileNotFoundError | test\_set\_colang\_config\_error |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Módulo: Nemo Core

| Función | \_\_init\_\_ | |
| --- | --- | --- |
| Descripción | Inicializar e Nemo Core | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Variables internas esperadas:  - nemo\_core.db == mock\_vector\_store  -nemo\_core.nemo\_config == mock\_nemo\_config  - nemo\_core.LLM == mock\_llm  - nemo\_core.rails is not None | test\_init |
| No Válido: N/A | N/A | N/A |

| Función | register\_action | |
| --- | --- | --- |
| Descripción | Registra una acción en el sistema de rieles (rails) | |
| Inputs | Expected Output | Test ID |
| Válido: {mock\_callback, test\_action} | nemo\_core.rails.register\_action es llamado una vez | test\_register\_action |
| No Válido: N/A | N/A | N/A |

| Función | register\_actions | |
| --- | --- | --- |
| Descripción | Registra múltiples acciones en el sistema. | |
| Inputs | Expected Output | Test ID |
| Válido: {mock\_callback, test\_action, array de are\_questions} | nemo\_core.rails.register\_action es llamado tres veces | test\_register\_actions |
| No Válido: N/A | N/A | N/A |

| Función | \_set\_context | |
| --- | --- | --- |
| Descripción | Establece el contexto para el procesamiento. | |
| Inputs | Expected Output | Test ID |
| Válido: {mock\_context} | El resultado contiene "global\_context", "client\_context", y "other\_context" | test\_set\_context |
| No Válido: N/A | N/A | N/A |

| Función | \_get\_complete\_args | |
| --- | --- | --- |
| Descripción | Obtiene los argumentos completos para diferentes tipos de contenido.  Inputs válidos:  - "question": Para obtener argumentos de pregunta.  - "answer": Para obtener argumentos de respuesta.  - "example": Para obtener argumentos de ejemplo. | |
| Inputs | Expected Output | Test ID |
| Válido: {“question”} | MOCK: {"content": {"question": True, "answer": False, "example": False}} | test\_get\_complete\_args |
| Válido: {“answer”} | MOCK: {"content": {"question": False, "answer": True, "example": False}} | test\_get\_complete\_args |
| Válido: {“example”} | MOCK: {"content": {"question": False, "answer": False, "example": True}} | test\_get\_complete\_args |

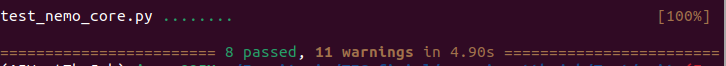
| Función | processCall | |
| --- | --- | --- |
| Descripción | Procesa una llamada al sistema | |
| Inputs | Expected Output | Test ID |
| Válido: {mock\_context} | MOCK: {"content": "Generated response"} | test\_processCall |
| No Válido: N/A | N/A | N/A |

| Función | update\_db | |
| --- | --- | --- |
| Descripción | Actualiza la base de datos con un nuevo documento | |
| Inputs | Expected Output | Test ID |
| Válido: {mock\_documentt} | - nemo\_core.db.load\_and\_embed es llamado con "test\_document.txt"  - os.remove es llamado con "test\_document.txt" | test\_update\_db |
| No Válido: N/A | N/A | N/A |

| Función | \_get\_context | |
| --- | --- | --- |
| Descripción | Obtiene el contexto para un texto dado | |
| Inputs | Expected Output | Test ID |
| Válido: {mock\_textt} | MOCK: ("global", "client", "other") | test\_get\_context |
| No Válido: N/A | N/A | N/A |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:

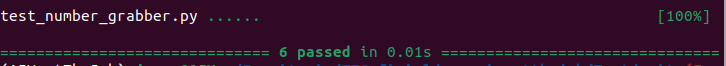


### Módulo: NumberGrabber

| Función | grab\_number | |
| --- | --- | --- |
| Descripción | Extrae el primer número entero de una cadena de texto | |
| Inputs | Expected Output | Test ID |
| Válido: {“The answer is 42"} | 42 | test\_grab\_un\_numero |
| Válido: {"There are 3 apples and 5 oranges"} | 3 | test\_grab\_solo\_primer\_numero |
| Válido: {There are no numbers here} | 0 | test\_grab\_sin\_numeros |
| Válido {"The temperature is -5 degrees"} | 0 | test\_grab\_number\_negativo |
| Válido {"The price is 10.99 dollars"} | 0 | test\_grab\_number\_decimal |
| Válido {“”} | 0 | test\_grab\_number\_string\_vacio |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:

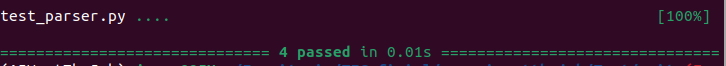


### Módulo: Parser

| Función | parser | |
| --- | --- | --- |
| Descripción | Analiza los argumentos de línea de comando. | |
| Inputs | Expected Output | Test ID |
| Válido: ['script\_name', 'test\_token'] | {'token': 'test\_token', 'model\_name': 'meta-llama/Llama-2-7b-chat-hf', 'cuantization': '4bit'} | test\_parser\_por\_defecto |
| Válido: ['script\_name', 'test\_token', '-n', 'custom\_model', '-c', '8bit'] | {'token': 'test\_token', 'model\_name': 'custom\_model', 'cuantization': '8bit'} | test\_parser\_valores\_custom |
| No Válido: [“script name”] | Lanza SystemExit | test\_parser\_sin\_token |
| No Válido: ['script\_name', 'test\_token', '--invalid\_arg', 'value'] | Lanza SystemExit | test\_parser\_argumento\_equivocado |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:

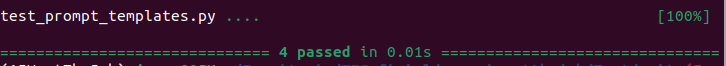


### Módulo: PromptTemplates

| Función | Verificación Atributos | |
| --- | --- | --- |
| Descripción | Verifica la existencia y tipo de los atributos de la clase PromptTemplates. | |
| Inputs | Expected Output | Test ID |
| Válido: {Vacío} | - PromptTemplates tiene el atributo question\_sysprompt\_template  - question\_sysprompt\_template es una cadena de texto | test\_entrevistador\_modo\_pregunta |
| Válido: {Vacío} | - PromptTemplates tiene el atributo give\_example\_output\_sysprompt\_template  - give\_example\_output\_sysprompt\_template es una cadena de texto | test\_entrevistador\_modo\_ejemplo |
| Válido: {Vacío} | - PromptTemplates tiene el atributo punctuate\_answer\_sysprompt\_template  - punctuate\_answer\_sysprompt\_template es una cadena de texto | test\_entrevistador\_modo\_evaluacion |
| Válido: {Vacío} | - PromptTemplates tiene el atributo phi1template  - phi1template es una cadena de texto | test\_template\_de\_phi |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Módulo: Proxy

| Función | \_\_init\_\_ | |
| --- | --- | --- |
| Descripción | Inicializa la clase Proxy. | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | Variables internas esperadas:  - proxy.database is not None  - proxy.llm is not None  - proxy.nemo\_system is not None  - proxy.cache is not None | test\_init |
| No Válido: N/A | N/A | N/A |

| Función | get\_data | |
| --- | --- | --- |
| Descripción | Inicializa la clase Proxy. | |
| Inputs | Expected Output | Test ID |
| Válido: {MOCK: "test", "question", "user1"} | MOCK: {"respuesta": "Generated question"} | test\_get\_data\_qeustion |
| Válido: {MOCK: "test", "example", "user1"} | MOCK: {"respuesta": "Example response"} | test\_get\_data\_example |
| Válido: {MOCK: "test", "answer", "user1"} | MOCK: {"respuesta": "Generated answer with score 8", "array": [8]} | test\_get\_data\_answer |

| Función | registrar\_usuario (mismo test para las dos clases) | |
| --- | --- | --- |
| Descripción | Registra un nuevo usuario en la base de datos | |
| Inputs | Expected Output | Test ID |
| Válido: {Usuario no existe} | MOCK: {"id": "user123"} | test\_registrar\_usuario |
| No Válido: {Usuario existe] | MOCK: None | test\_registrar\_usuario |

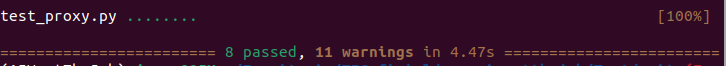
| Función | validar\_usuario (mismo test para las dos clases) | |
| --- | --- | --- |
| Descripción | Valida las credenciales de un usuario. | |
| Inputs | Expected Output | Test ID |
| Válido: {Usuario existe | MOCK: {"id": "user123"} | test\_validar\_usuario |
| No Válido: {Usuario no existe] | MOCK: None | test\_validar\_usuario |

| Función | insertar\_valor\_array | |
| --- | --- | --- |
| Descripción | Inserta un valor en el array de un usuario. | |
| Inputs | Expected Output | Test ID |
| Válido: {MOCK id y valor} | MOCK: {"array": [1, 2, 3]} | test\_insertar\_valor\_array |
| No Válido: Comprobado en BD (recibe un array por defecto que entra dentro del scope del test anterior) | N/A | N/A |

| Función | recuperar\_valores\_array | |
| --- | --- | --- |
| Descripción | Inserta un valor en el array de un usuario. | |
| Inputs | Expected Output | Test ID |
| Válido: {MOCK id} | MOCK: {"array": [1, 2, 3]} | test\_recuperar\_valores\_array |
| No Válido: N/A Comprobado en BD  (recibe un array por defecto que entra dentro del scope del test anterior) | N/A | N/A |

#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Módulo: VectoreStore

| Función | \_\_init\_\_ | |
| --- | --- | --- |
| Descripción | Inicializa la clase VectoreStore. | |
| Inputs | Expected Output | Test ID |
| Válido: {por defecto} | - instancia\_mockeada.client == mock\_chromadb\_client.return\_value  - instancia\_mockeada.embedding\_model is not None | test\_init |
| No Válido: N/A | N/A | N/A |

| Función | \_start\_embd\_model | |
| --- | --- | --- |
| Descripción | Inicia el modelo de embedding | |
| Inputs | Expected Output | Test ID |
| Válido: {"sentence-transformers/all-MiniLM-L6-v2"} | El modelo retornado no es None | test\_start\_embd\_model |
| No Válido: N/A | N/A | N/A |

| Función | load\_and\_embed | |
| --- | --- | --- |
| Descripción | Carga los documentos en la vectorstore | |
| Inputs | Expected Output | Test ID |
| Válido: {"document\_route"} | - mock\_loader\_instance.load es llamado una vez  - mock\_splitter\_instance.split\_documents es llamado una vez con content\_mock  - mock\_embed\_documents es llamado una vez con ["split\_content"] | test\_load\_and\_embed |
| No Válido: N/A | N/A | N/A |

| Función | get\_context | |
| --- | --- | --- |
| Descripción | Obtiene el contexto general, del cliente y de similitud. | |
| Inputs | Expected Output | Test ID |
| Válido: {"text"} | - general\_context == "context"  - client\_context == "context"  - similarity\_context == "question"  - \_get\_document es llamado dos veces con ("text", "general\_context") y ("text", "client\_context")  - \_get\_question es llamado una vez con ("text", "similarity\_context") | test\_get\_context |
| No Válido: N/A | N/A | N/A |

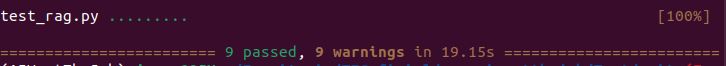
| Función | \_get\_document | |
| --- | --- | --- |
| Descripción | Obtiene el contexto general, del cliente y de similitud. | |
| Inputs | Expected Output | Test ID |
| Válido: {"text", “index\_name”} | MOCK: “document” | test\_get\_document |
| No Válido: {Execpción por error} | Execepción | test\_get\_document\_exception |

| Función | \_get\_question | |
| --- | --- | --- |
| Descripción | Obtiene una pregunta similar de una colección específica | |
| Inputs | Expected Output | Test ID |
| Válido: {MOCK: "text", “index\_name”, “similarity\_th”} | MOCK: None o “Question” en base al MOCK similarity\_th | test\_get\_question |
| No Válido: {Execpción por error} | Execepción | test\_get\_question\_exception |

| Función | add\_question | |
| --- | --- | --- |
| Descripción | Añade una pregunta a la colección de similitud | |
| Inputs | Expected Output | Test ID |
| Válido: {MOCK: “Context” y “Question} | - mock\_get\_collection es llamado una vez con "similarity\_context"  - mock\_embed\_query es llamado una vez con "%CONTEXT%context%QUESTION%question"  - mock\_collection.add es llamado una vez con los ids y embeddings correctos | test\_add\_question |
| No Válido: N/A (hardcodeado) | N/A | N/A |

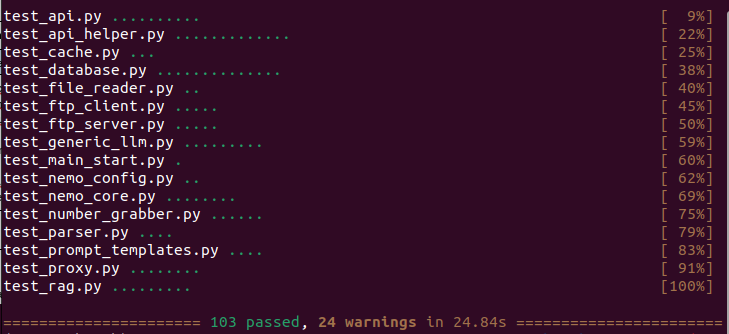
#### Resultados

Los resultados de ejecutar los tests para este módulo son los siguientes:



### Resultados Generales

Los test unitarios se pasan con un 100% de ratio de paso. Se adjunta a continuación los resultados.



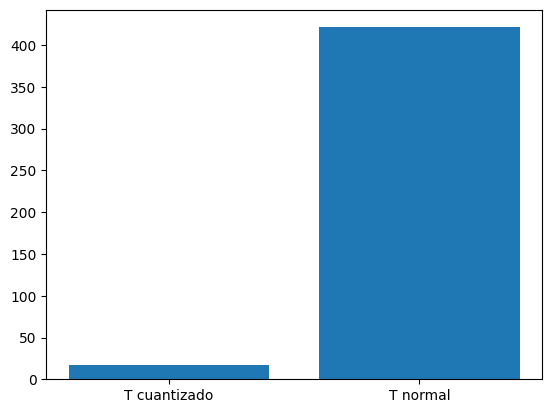
## Pruebas de integración

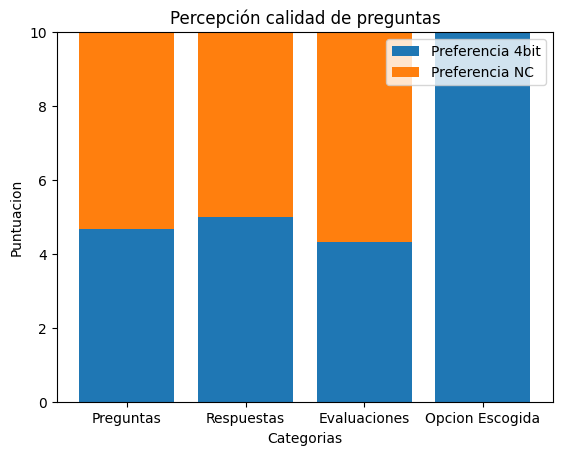
En las pruebas de integración se han realizado notebooks en los que se prueba la integración de los módulos entre śi, probando cambios de flujo estandarizados.

En el apartado de Tests/integration se ofrecen los siguientes tests:

* Tests de comunicación entre la API y la interfaz del frontend
* Tests de comunicación entre el servidor y el cliente ftp
* Tests de funcionamiento de la base de datos
* Tests de funcionamiento del LLM genérico
* Tests de funcionamiento del Gestor de comportamientos
* Tests de funcionamiento del Proxy
* Tests de funcionamiento del RAG
* Tests de generación de inferencia con el entrevistador

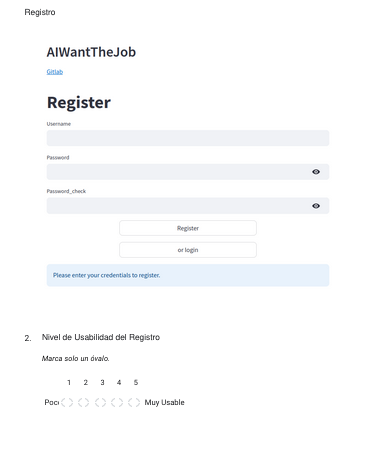
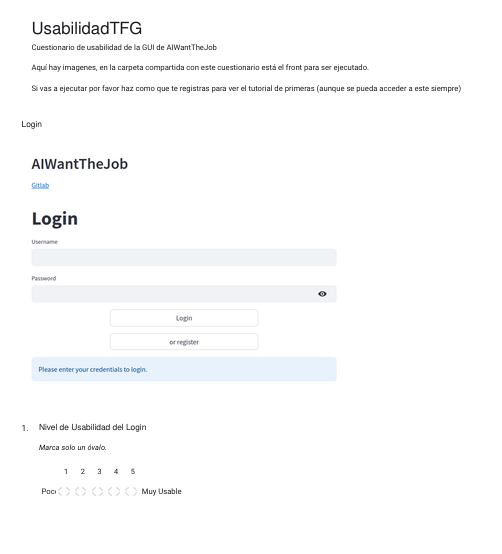
A mayores, se hace una prueba de generación de muchas respuestas del bot del que se ha evaluado tanto el tiempo de respuestas como la calidad (también es parte de usabilidad). Los resultados se pueden ver en las siguientes gráficas. Para ver el excel de preguntas ir a Tests/Integration en el repositorio.

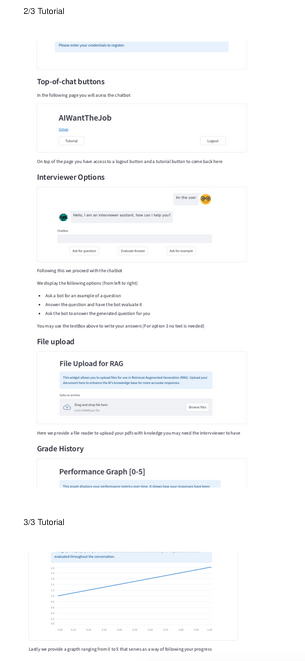
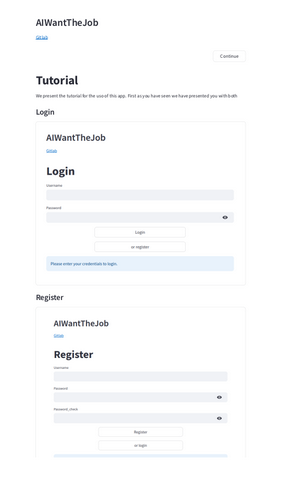


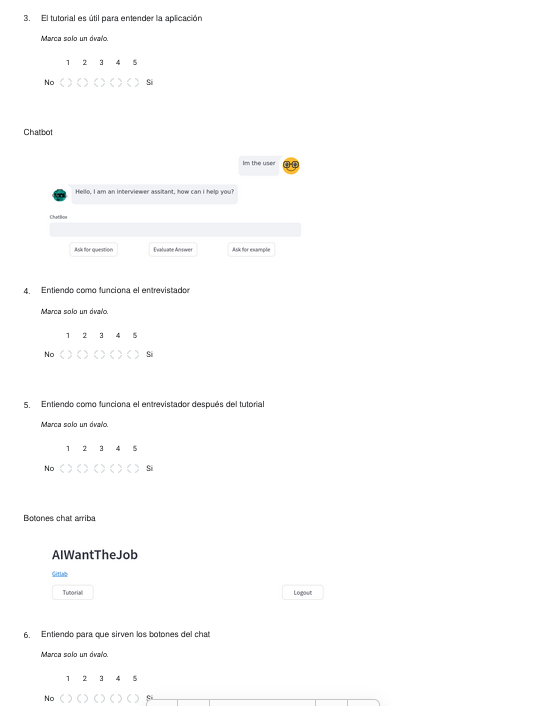


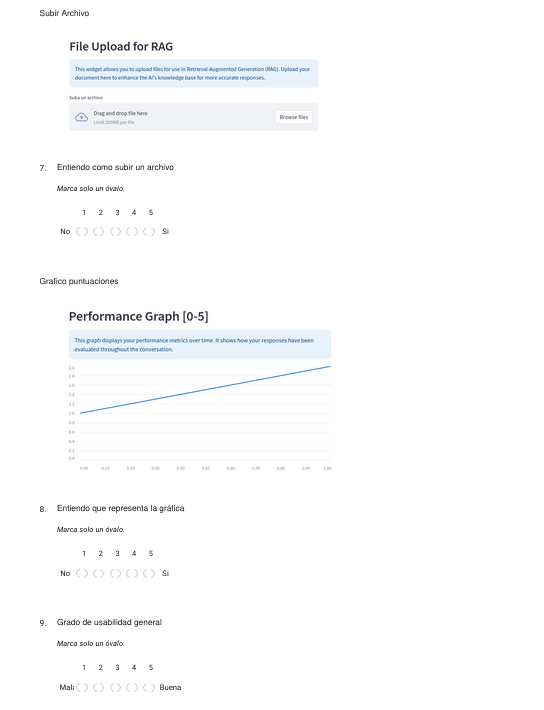
## Pruebas de usabilidad

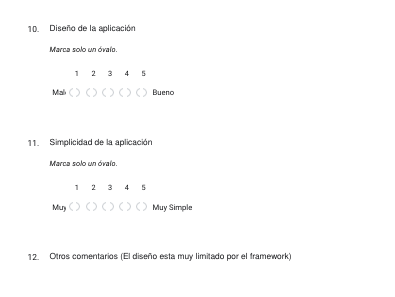
El primer criterio del que vamos a evaluar la usabilidad es de la GUI. Para ello hemos construido un mock en el apartado Tests/usability del repositorio. A partir de ese mock construimos un cuestionario (para responder con imágenes y con el mock). El enlace: <https://docs.google.com/forms/d/e/1FAIpQLSemyPiMTV0eUEMfbmuZ1VRNAyproW8tO3blx4TaVicVJqw5cA/viewform?usp=sf_link>











Las respuestas a este cuestionario fueron las siguientes:

Gráfico de respuestas de formularios. Título de la pregunta: Nivel de Usabilidad del Login
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: Nivel de Usabilidad del Registro
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: El tutorial es útil para entender la aplicación
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: Entiendo como funciona el entrevistador
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: Entiendo como funciona el entrevistador después del tutorial
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: Entiendo para que sirven los botones del chat
. Número de respuestas: 3 respuestas.

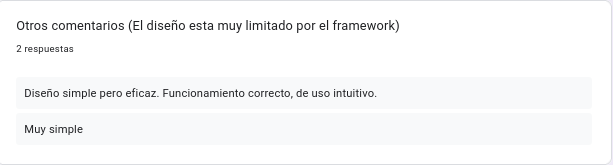
Gráfico de respuestas de formularios. Título de la pregunta: Entiendo como subir un archivo
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: Entiendo que representa la gráfica
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: Grado de usabilidad general
. Número de respuestas: 3 respuestas.

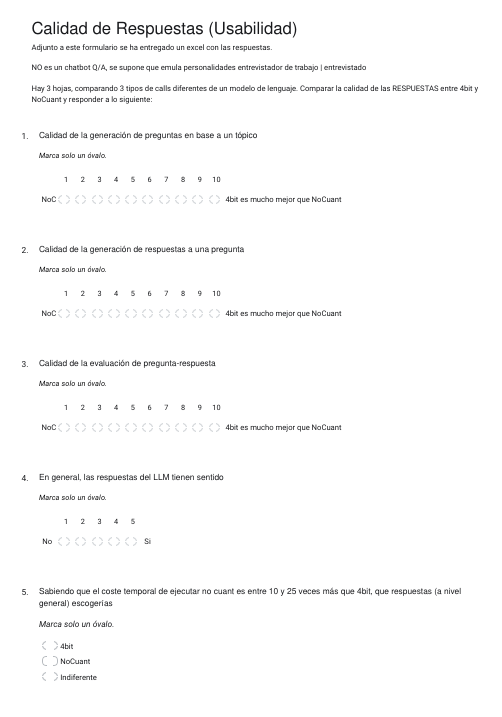
Gráfico de respuestas de formularios. Título de la pregunta: Diseño de la aplicación
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: Simplicidad de la aplicación
. Número de respuestas: 3 respuestas.



Luego realizamos inferencia del modelo con y sin cuantizar (como ya se explicó en el apartado de integración) y realizamos un cuestionario comprobando la calidad de las respuestas para los usuarios. El enlace al cuestionario es el siguiente: <https://docs.google.com/forms/d/e/1FAIpQLSfagpm9ZYqlZPBkdShZT-6ay5jc19Z0bGT4__0uWdKGMuoiiA/viewform?usp=sf_link>

Este cuestionario fue el siguiente:



Las respuestas a este cuestionario fueron las siguientes:

Gráfico de respuestas de formularios. Título de la pregunta: Calidad de la generación de preguntas en base a un tópico
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: Calidad de la generación de respuestas a una pregunta
. Número de respuestas: 3 respuestas.

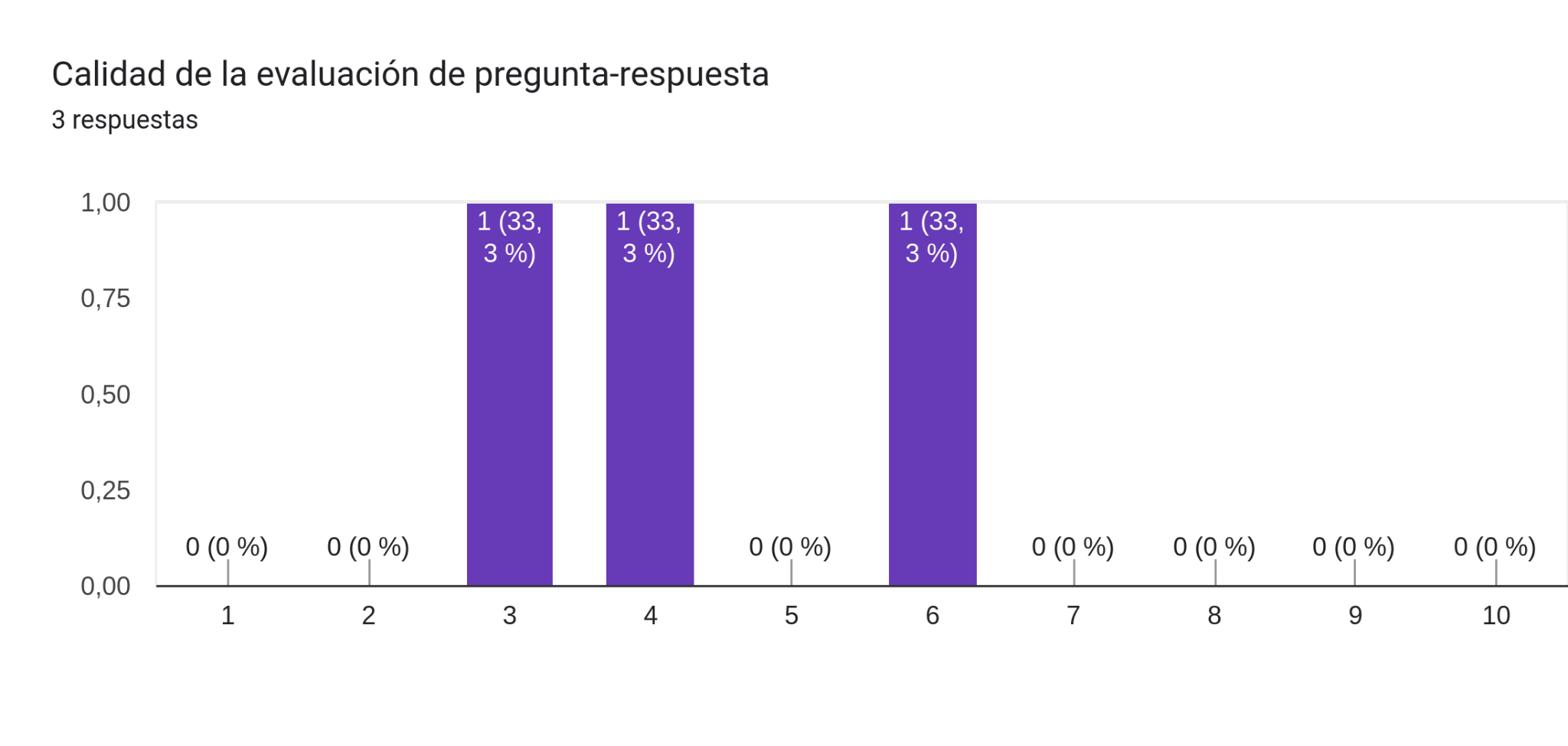


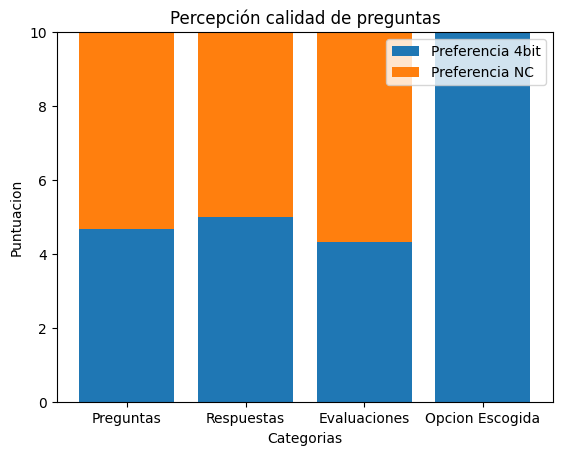
Gráfico de respuestas de formularios. Título de la pregunta: En general, las respuestas del LLM tienen sentido
. Número de respuestas: 3 respuestas.

Gráfico de respuestas de formularios. Título de la pregunta: Sabiendo que el coste temporal de ejecutar no cuant es entre 10 y 25 veces más que 4bit, que respuestas (a nivel general) escogerías
. Número de respuestas: 3 respuestas.

### Conclusiones del apartado

De la parte de diseño se concluye que la GUI es muy usable y sencilla de diseño (esto es tanto algo bueno como algo malo).

Por otro lado, de los resultados de la encuesta se puede resumir la siguiente gráfica, con la que podemos ver que no merece la pena usar un modelo más grande ya que el pequeño funciona muy bien.



## Pruebas end-to-end

Las pruebas de extremo a extremo se verifican con la realización de los flujos completos de trabajo del usuario. En este caso, como se presenta una aplicación que implementa todos sus objetivos y requisitos, las pruebas de extremo a extremo se pasan por el funcionamiento de dicha aplicación

## Pruebas de despliegue

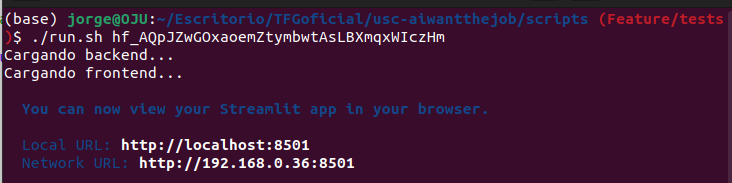
Para probar el despliegue realizamos dicho despliegue a través de todos los métodos de despliegue planteados:

### Despliegue manual

El funcionamiento del despliegue manual se auto verifica porque la aplicación fue desarrollada así y es el método que use como desarrollador por defecto.

### Despliegue conda

Podemos ver como ejecutando el script de run (conda en el background gracias al comando source) el código funciona. De esta manera ofrecemos el entorno virtualizado. El video de demo en el repositorio está realizado ejecutando el proyecto en conda.



## Conclusión

Hemos pasado todas las pruebas planteadas. De esta manera cumplimos el criterio de aceptación A-001 y se determinan las pruebas realizadas como pasadas.