

Proyecto Espacio de Observación de Inteligencia Artificial en Español

Ámbito 1.3 Aplicación Web EvaLL 2.0

Informe Técnico Año 2

**Enrique Amigó, Jorge Carrillo-de-Albornoz, Andrés Fernández, Julio Gonzalo,
Guillermo Marco, Roser Morante, Jacobo Pedrosa, Laura Plaza, Eva Sánchez**

Natural Language Processing and Information Retrieval Group, UNED

Autor de contacto: Julio Gonzalo - julio@lsi.uned.es

Resumen

En este informe se presenta la versión 2 de la aplicación web EvaLL 2.0, que se ha desarrollado en la UNED en el marco del proyecto del Espacio de Observación de Inteligencia Artificial en Español, en concreto “Ámbito 1 Estado del arte comparado”, “Actividad 1.3 Aplicación web EvaLL 2.0”.

1. Introducción

En este informe se presenta la versión 2 del portal web de evaluación EvALL 2.0 para sistemas de información. El portal ha sido desarrollado por el grupo de investigación en Procesamiento de Lenguaje Natural y Recuperación de Información de la UNED, en el marco del proyecto del Espacio de Observación de Inteligencia Artificial en Español, en concreto Ámbito 1 Estado del arte comparado, Actividad 1.3 - Aplicación web EvALL 2.0.

EvALL 2.0 (Evaluate ALL 2.0), es una herramienta que permitirá evaluar sistemas de información sobre un conjunto extenso de métricas que abarcan multitud de contextos de evaluación, entre los que se incluyen clasificación, ranking, o clustering, así como datos con y sin desacuerdo, entre otros. La evaluación en EvALL 2.0 es entendida como un proceso que evoluciona con el tiempo. Es por esto que EvALL está diseñada sobre los siguientes conceptos: (i) **persistencia**, el usuario puede almacenar evaluaciones, así como recuperar evaluaciones pasadas; (ii) **replicabilidad**, todas las evaluaciones son realizadas siguiendo la misma metodología, por lo que son estrictamente comparables; (iii) **efectividad**, todas las métricas se engloban bajo la teoría de la medida ([Amigó et al., 2023](#); [Amigó and Mizzaro, 2020](#)), y han sido doblemente implementadas y comparadas. El objetivo final del desarrollo de EvALL 2.0 es proporcionar a la comunidad científica y a los desarrolladores de herramientas de Procesamiento de Lenguaje Natural (PLN) una aplicación de evaluación de referencia, con un carácter didáctico, y con una mínima curva de aprendizaje.

Durante esta segunda anualidad de ODESIA, se ha realizado un trabajo que resulta esencial en el desarrollo de toda aplicación software de cierta envergadura, como es el diseño y desarrollo de un primer prototipo amigable y usable. El prototipo desarrollado facilita a la comunidad científica y a la industria uno de los procesos más importantes para el avance de la investigación, como es la evaluación. Además, EvALL 2.0 lo hace de una forma transparente y didáctica para el usuario, reconociendo al proceso de evaluación su papel clave en el avance de la ciencia. En concreto, durante este año se ha rediseñado la apariencia de la aplicación, y se ha hecho un gran esfuerzo en mejorar el diseño y desarrollo del dashboard, el centro neurálgico de EvALL 2.0. Este desarrollo incluye nuevas formas de visualizar y analizar los resultados, tanto textual (mediante los informes de PyEvALL), como visual (mediante gráficas generadas dinámicamente). Además, se han incluido los flujos de gestión de evaluaciones, la evaluación contra repositorio y se ha desarrollado el repositorio EvALL 2.0. Respecto a la librería de evaluación PyEvALL, se han incluido tres nuevos contextos de evaluación, y se han implementado 10 métricas nuevas asociadas a dichos contextos y a otros ya existentes. Por último, se ha mejorado el tratamiento de errores en los

formatos de predicciones y *gold standard*, así como el análisis de las pre-condiciones de las métricas que lo precisan.

1.1. Motivación

La evaluación de los resultados obtenidos en una experimentación es crucial en toda investigación científica. Sin embargo, no es raro encontrar metodologías de evaluación débiles o métricas poco apropiadas al problema abordado. No es fácil elegir, utilizar e interpretar adecuadamente una métrica de evaluación. Dado que la comparación con el estado del arte es esencial, los investigadores tienden a centrarse en metodologías y métricas de evaluación populares, aunque el estado del arte en materia de evaluación haya avanzado. Como resultado, las métricas con propiedades formales y empíricas preferibles suelen descartarse en favor de métricas heredadas que garantizan la comparabilidad retrospectiva. Por otro lado, los investigadores tienden a centrarse en el desarrollo de sistemas y dedican poco tiempo a seleccionar y comprender las métricas de evaluación. Una vez más, recurrir a métricas ampliamente adoptadas es una alternativa segura, al precio de una interpretación inadecuada (y a veces engañosa) de los resultados experimentales.

Otro problema común es el uso frecuente de métricas de evaluación incluidas en paquetes de aprendizaje automático como Scikit-learn¹ o Keras². En estos paquetes, la evaluación es abordada de forma transparente al usuario, sin identificar posibles errores que pueden ocurrir en toda evaluación, y que se han de abordar para obtener resultados precisos (instancias duplicadas, elementos que no existen en el *gold standard*, clases objetivo incorrectas, etc.). Por el contrario, estas comprobaciones quedan relegadas al usuario al requerir para su ejecución simplemente vectores de una dimensión, lo que se traduce en una mayor probabilidad de resultados inexactos, especialmente entre los usuarios menos expertos. Por contra, el análisis de errores sí es un aspecto central en otras herramientas de evaluación como *trec eval*³, cuyo objetivo principal es la evaluación de sistemas de información. El principal problema de este tipo de herramientas es que se centran solo en un contexto de evaluación (por ejemplo, *trec eval* se centra en el contexto de ranking), y por lo general, en un conjunto reducido de métricas.

En resumen, EvALL 2.0 tratar de abordar los siguientes problemas detectados en el estado del arte de la evaluación de sistemas de información:

- Escasa, o nula, disponibilidad de *frameworks* de evaluación que incluyan varios contextos de evaluación, como por ejemplo: clasificación mono-label, clasificación multi-label, clasificación jerárquica, ranking o clustering, entre otros.
- Limitada capacidad de elección de métricas para cada contexto de evaluación, centrándose, en general, en las métricas más populares.
- Escaso, o nulo, proceso de detección de errores en los conjuntos de datos de predicciones y los datos del *gold standard* (identificación de instancias duplicadas, clases incorrectas o campos vacíos, entre otros).
- Escasa, o nula, evaluación de las pre-condiciones de las métricas, determinando en cada caso la viabilidad de su ejecución.
- Evaluaciones no persistes en el tiempo, dado que las mayoría de herramientas producen los resultados de forma textual en consola.
- Escasa, o nula, disponibilidad de diferentes formatos de informes de evaluación.

¹<https://scikit-learn.org/stable/>

²<https://keras.io/>

³https://github.com/usnistgov/trec_eval

1.2. Objetivos específicos

De acuerdo a las limitaciones existentes en el estado del arte de las herramientas y librerías de evaluación de sistemas de información, en este trabajo nos hemos fijado los siguientes objetivos en el desarrollo de EvALL 2.0:

- **Universalidad:** la universalidad de EvALL 2.0 vendrá garantizada gracias al amplio conjunto de métricas incluidas en la herramienta, y que abarcan un gran abanico de contextos de evaluación: clasificación mono-label, clasificación multi-label, clasificación jerárquica, clasificación ordinal, ranking, ranking con diversidad, o clustering, entre otros. En concreto, se estima que mediante la interfaz amigable de EvALL 2.0 el usuario podrá ejecutar más de 40 métricas, lo que dota a la herramienta de una universalidad actualmente no disponible en ninguna herramienta de evaluación.
- **Generalización:** la generalización viene determinada por el uso de un formato estandarizado de entrada que permita al usuario ejecutar métricas de contextos tan diversos como ranking o clustering con un solo archivo de predicciones y gold standard. El formato EvALL, a su vez, hace uso del formato *json*, un formato altamente extendido y conocido por la comunidad científica. Por último, EvALL 2.0 dispondrá de multitud de *wrappers* que automáticamente convertirán los archivos de entrada a formato EvALL permitiendo así la generalización deseada.
- **Precisión:** la precisión de la evaluación en EvALL 2.0 viene garantizada gracias a los siguientes puntos:
 - **Análisis de archivos de entrada:** en cada evaluación se realizará un análisis de los archivos de entrada, tanto de las predicciones como del *gold standard*, para la detección de posibles errores e inconsistencias en los mismos. En caso de detectar errores, EvALL impedirá la evaluación, si estos son producidos en el *gold standard*, e informará y continuará, si estos son producidos en las predicciones.
 - **Comprobación de pre-condiciones:** para cada métrica, siempre y cuando la misma lo requiera, se determinará si es aplicable, o no, de acuerdo a sus pre-condiciones sobre los datos de entrada.
 - **Doble implementación:** todas y cada una de las métricas incluidas en EvALL 2.0 son comparadas con otra implementación independiente y evaluadas con diferentes casos de prueba diseñados a tal propósito.
- **Persistencia:** EvALL 2.0 contará con un repositorio donde los usuarios podrán almacenar, así como recuperar en el futuro, sus evaluaciones de forma privada. Así mismo, los usuarios podrán publicar tareas y resultados asociados a estas de forma pública, para que toda la comunidad científica pueda consultarlos.
- **Amigable:** esta nueva versión persigue la idea de una herramienta apta tanto para expertos como principiantes en el área de la evaluación de sistemas de información. Por ello, EvALL 2.0 debe ajustarse a los patrones actuales de navegación en la que, con unos pocos pasos, cualquier usuario, investigador o desarrollador, pueda realizar una evaluación transparente para él.
- **Didáctica:** otro de los objetivos deseados de la aplicación es la progresiva concienciación de la importancia de la evaluación de un sistema de información y de las diferentes perspectivas desde la que se puede abordar. Para ello, EvALL 2.0 dispondrá de diferentes opciones de evaluación ya pre-configuradas, así como distintos tipos de informes con explicaciones sobre las métricas, sus formulas matemáticas y su interpretación.

2. Diseño y arquitectura de EvALL 2.0

En la sección anterior, se han descrito la motivación y los objetivos principales para el desarrollo de la aplicación web EvALL 2.0. Como ya se ha mencionado, el objetivo principal de EvALL 2.0 es convertirse en una aplicación de referencia en el área de evaluación de sistemas de información. En la presente sección, se describen en detalle el diseño y la arquitectura propuestos para el desarrollo de la aplicación (que se extiende durante toda la duración del proyecto). En primer lugar, se describen los potenciales perfiles de usuario a los que va dirigida. A continuación, se analizan los casos de uso, o flujos de ejecución, con los que contará la aplicación EvALL 2.0, seguido de la extracción de requisitos necesarios para alcanzar las funcionalidades deseadas. Extraídos los requisitos, se analizan las tecnologías existentes y se propone una arquitectura basada en la tecnología más adecuada para nuestro caso de uso. Finalmente, se describe el diseño de la arquitectura propuesta, así como la composición de cada una de sus capas y los roles de usuario.

2.1. Perfiles de usuario

Para conseguir el objetivo marcado, la aplicación EvALL 2.0 debe ser de utilidad e interés para un gran número de usuarios con perfiles muy diversos. En concreto, se han identificado los siguientes perfiles a los que EvALL 2.0 va dirigida:

- **Investigadores:** son el principal grupo de usuarios a los que va dirigida la aplicación, dado que la evaluación es un proceso fundamental de su trabajo diario. Por ello, disponer de una herramienta de evaluación de referencia con un amplio conjunto de contextos de evaluación, y métricas asociadas a cada contexto, supone un gran beneficio. Dentro del grupo de investigadores se pueden diferenciar, a su vez, tres perfiles diferentes:
 - **Expertos en evaluación:** aquellos que, por su área de investigación, poseen mayor experiencia y conocimiento de las distintas opciones de evaluación de sistemas de información. EvALL 2.0 puede beneficiar a este grupo al dotarles de una herramienta de fácil uso en la que comparar y evaluar resultados desde diferentes contextos de evaluación y métricas.
 - **Principiantes en evaluación:** investigadores cuyo área de conocimiento no es la evaluación de sistemas y relegan esa parte a herramientas de confianza. En este sentido, EvALL 2.0 pretende no solo ser una herramienta fiable y de uso sencillo, sino un elemento didáctico que sirva a sus usuarios para entender mejor los procesos de evaluación.
 - **Organizadores de campañas de evaluación:** aquellos investigadores que organicen una campaña de evaluación disponen en EvALL 2.0 de una herramienta fiable y persistente para evaluar los resultados de los participantes. Así mismo, EvALL 2.0 garantiza que todos los sistemas son evaluados bajo la misma metodología, pudiendo incluso comparar entre distintas campañas de evaluación si se dan los parámetros adecuados.
- **Desarrolladores:** tanto a investigadores como a desarrolladores de empresas o administraciones, EvALL 2.0 puede ofrecer un *framework* de evaluación fácil de usar, con un gran abanico de opciones, donde dichos usuarios no necesariamente deben conocer aspectos relevantes de la evaluación de sistemas de información. Para ello, al finalizar todo el proyecto se proporcionará a la comunidad (mediante licencia de código abierto) todo el desarrollo, especialmente la librería de evaluación PyEvALL que se describirá en las siguientes secciones.
- **Administraciones y empresas:** EvALL 2.0 puede ayudar a administraciones y empresas a validar la eficiencia de los posibles sistemas de información que deseen adquirir. Por ejemplo, hospitales que deseen comprar un sistema automático de clasificación de códigos CIE pueden solicitar que la empresa realice las pruebas pertinentes sobre la plataforma EvALL 2.0, garantizando así su fiabilidad.
- **Profesores:** EvALL 2.0 puede ser un referente para profesores de disciplinas como informática, telecomunicaciones, etc., en donde iniciar al alumnado en el área de evaluación de sistemas de información.

- **Alumnos:** alumnos de disciplinas relacionadas con informática, telecomunicaciones, etc., pueden estar interesados en utilizar EvALL 2.0 para validar sus algoritmos, a la par que adquirir más conocimientos sobre distintas opciones y contextos de evaluación.

2.2. Casos de uso

Dado el gran abanico de perfiles de usuario a los que EvALL 2.0 desea servir, se hace necesario dotar a la aplicación con un conjunto de casos de uso, o flujos de ejecución, muy diverso. Esto ayudará, además, a conseguir el objetivo de convertir a EvALL 2.0 en una herramienta de evaluación de referencia y fácil uso. Sin embargo, la complejidad de la aplicación irá escalando según se vayan añadiendo nuevas funcionalidades, por lo que un correcto diseño que permita dicha flexibilidad se hace imprescindible.

Así mismo, la ejecución de aplicaciones de evaluación puede llegar a ser un proceso costoso en tiempo de procesamiento y recursos, pudiendo llegar a colapsar el sistema. Por todo ello, y como requisito necesario para tener un correcto control de la aplicación y los distintos flujos de ejecución, es imprescindible que EvALL 2.0 sea una aplicación accesible únicamente con registro de usuario.

Atendiendo a esto, se proponen los siguientes flujos de ejecución de la aplicación EvALL 2.0, y que guiarán el desarrollo de la aplicación durante los tres años de ejecución del proyecto:

- **Flujo registro de usuario:** mediante este flujo, un usuario no registrado previamente podrá crear una cuenta en la aplicación de EvALL 2.0. Para ello, deberá proporcionar cierta información, como por ejemplo, datos de identificación y correo electrónico, así como aceptar los términos de uso de la aplicación.
- **Flujo inicio de sesión:** mediante este flujo de ejecución, un usuario previamente registrado podrá iniciar sesión en EvALL 2.0 y acceder así a las diferentes opciones de evaluación o a la gestión de su perfil.
- **Flujo gestión de perfil:** dado que EvALL 2.0 es una aplicación en la que el registro de usuario es necesario, se debe dotar a los usuarios de una interfaz en la que gestionar su perfil y la información asociada al mismo.
- **Flujo evaluación proporcionando el *gold standard*:** mediante este flujo de evaluación, un usuario registrado podrá realizar la evaluación de sus modelos de predicción frente a un *gold standard*. Para ello, EvALL 2.0 proporcionará una interfaz amigable mediante la cual, en solo cuatro pasos, cualquier usuario, independientemente de su perfil y experiencia, pueda evaluar sus modelos. En concreto, el usuario solo tendrá que: (i) subir el archivo *gold standard*; (ii) subir uno o varios archivos con las predicciones de sus modelos; (iii) seleccionar las métricas deseadas; (iv) pulsar el botón *Evaluar*. Nótese que este flujo puede ser de gran interés tanto para investigadores como para desarrolladores. Así mismo, profesores y alumnos pueden encontrar en este flujo una manera sencilla de introducirse en el área de la evaluación de sistemas de información.
- **Flujo evaluación con *gold standard* desde repositorio:** este flujo de evaluación pretende simplificar aún más la evaluación mediante la ejecución de la misma contra un *gold standard* asociado a una tarea ya almacenada en el repositorio de EvALL 2.0. De esta forma, la ejecución de una evaluación consta de los siguientes pasos: (i) seleccionar la tarea del repositorio; (ii) subir uno o varios archivos con predicciones; (iii) seleccionar las métricas deseadas; (iv) pulsar el botón *Evaluar ¿Enviar o evaluar?*. Este flujo, unido al flujo de publicar resultados, permitirá, así mismo, determinar la eficiencia del estado del arte respecto a una tarea almacenada en el repositorio. De esta manera, investigadores de todo el mundo podrán comparar sus sistemas bajo las mismas condiciones de evaluación. Este flujo puede ser de especial interés para investigadores que quieran publicitar un trabajo publicado en alguna conferencia o revista, así como empresas que quieran demostrar la efectividad de sus sistemas para resolver ciertas tareas.
- **Flujo organización de campañas de evaluación:** mediante este flujo, EvALL 2.0 ofrece soporte a los organizadores de campañas de evaluación. Este flujo pretende eliminar los tediosos procesos

de: (i) recopilación de datos de los participantes, tanto archivos de predicciones como información de participación; (ii) limpieza y detección de errores en los archivos enviados; (iii) generación de una evaluación fiable y comparable; (iv) generación de un informe final de evaluación. Para el correcto funcionamiento de este flujo, se dispondrá de dos roles claramente diferenciados: el rol de organizador, que inicia una campaña de evaluación y tiene acceso a toda la información de la tarea; y el rol de participante, que simplemente tiene acceso a los datos de entrenamiento y puede realizar el envío de las predicciones sobre el conjunto de prueba. Este flujo está fundamentalmente orientado a investigadores, que suelen ser los organizadores de campañas de evaluación. Por otro lado, tanto empresas como desarrolladores u otros investigadores pueden beneficiarse mediante el rol de participante.

- **Flujo publicar resultados:** mediante este flujo, un usuario puede publicar los resultados asociados a una evaluación y un archivo de predicciones. La publicación de los resultados implica la aceptación de publicación de ciertos detalles sobre el modelo publicado de cara a su mayor comprensión por parte de la comunidad. Nótese que los resultados de una evaluación solo pueden ser publicados si el *gold standard* de la tarea se encuentra en el repositorio de EvALL 2.0. Este puede ser un mecanismo muy útil, tanto para investigadores como empresas, para dar a conocer el estado del arte de sus aproximaciones a la comunidad científica, o a posibles clientes, con la garantía de eficiencia y calidad de EvALL 2.0.
- **Flujo publicar *gold standard*:** este flujo permite a los usuarios publicar un *gold standard* asociado a una tarea específica. Este proceso requiere verificación por parte de un administrador, dado su posible impacto en la aplicación y la comunidad científica, por lo que se realizará mediante un proceso en dos pasos. Es decir, el usuario enviará el *gold standard* para su publicación, así como todos los datos necesarios sobre el mismo y la aceptación de los términos de uso. Una vez enviada la información, un administrador de la aplicación EvALL 2.0 deberá aceptar la solicitud. Es importante tener en cuenta que la información almacenada en el repositorio es simplemente la necesaria para realizar la evaluación, es decir, las etiquetas de referencia, relegando la carga excesiva de almacenamiento de datos o procesamiento de sistemas que harían a este sistema inviable. Los investigadores son los principales usuarios de esta funcionalidad, ya que, a día de hoy, son los perfiles que hacen públicos los conjuntos de datos desarrollados.
- **Flujo consultar el repositorio:** mediante este flujo, cualquier usuario puede consultar el repositorio de EvALL 2.0, tanto tareas como sistemas asociados y sus resultados, pudiendo comparar el estado del arte actual para una determinada tarea. Este flujo está pensado para todo tipo de perfiles de usuarios, desde un investigador que desea conocer el estado del arte, a una empresa o administración que desea informarse a la hora de adquirir un producto en una tarea específica.
- **Flujo gestión de evaluaciones y *gold standards*:** por último, mediante este flujo cualquier usuario puede consultar y eliminar las evaluaciones, tanto públicas como privadas, y los *gold standards* subidos al repositorio por el usuario.

2.3. Requisitos

Como se ha descrito previamente, uno de los requisitos necesarios para el correcto control del funcionamiento de la aplicación EvALL 2.0 es que esté **orientada a usuarios registrados**, evitando así posibles problemas de recursos o mala praxis, a la par que permitiendo el almacenamiento de datos en repositorio. Los procesos de registro de usuario e inicio de sesión son procesos sensibles que, desde el punto de vista de la seguridad, conviene analizar en detalle. Por ello, lo más sencillo y frecuente es relegar este tipo de funcionalidades a sistemas de código abierto en donde las posibles brechas de seguridad son analizadas globalmente por la comunidad y resueltas en constantes actualizaciones.

Otro aspecto fundamental en EvALL 2.0 es el **carácter amigable y facilidad de uso** con los que se quiere dotar a la aplicación. Por ello, un requisito imprescindible es la creación de interfaces sencillas en

donde el usuario pueda visualizar, de una sola vez, todos los elementos necesarios para la realización de las evaluaciones. Para este propósito, se van a utilizar *dashboards*, adaptándolos en su caso a cada flujo de evaluación. Así mismo, la ejecución de una evaluación debe ser sencilla y realizable en pocos pasos, por lo que la lógica de cada flujo tiene que reducir al máximo las interacciones de usuario. Así mismo, debe permitir a usuarios más experimentados opciones más complejas como modificar parámetros de las métricas, comparativas entre resultados, etc. Por último, la unificación del formato de entrada para todos los contextos de evaluación, así como el uso de *wrappers* para otros formatos, limitará la barrera de entrada a usuarios principiantes. Todo esto, a su vez, debe ser acompañado de un completo manual de ayuda y FAQ.

Para que EvALL 2.0 pueda llegar a convertirse en un *framework* de evaluación de referencia debe estar diseñada sobre la idea de una **herramienta de propósito general y precisa**. Para ello, como ya se ha comentado, EvALL 2.0 debe incluir diferentes contextos de evaluación (clasificación mono-label y multi-label, clasificación jerárquica, ranking, clustering, clasificación ordinal, etc.), así como un gran abanico de métricas asociadas a cada contexto. Además, todas las métricas incluidas están basadas en la teoría de la medida, así como evaluadas con diferentes casos de prueba y comparadas frente a otras implementaciones de las mismas.

La evaluación es algo que va evolucionando con el tiempo, así como el estado del arte de los modelos evaluados. El poder disponer en un entorno, mediante un solo clic, de todas las evaluaciones realizadas, compararlas entre ellas, e incluso añadir nuevas, supone un claro beneficio para la comunidad. Además, la posibilidad de publicar tareas, así como resultados de predicciones asociadas a cada tarea, permite dotar a los usuarios de un marco de comparación entre sistemas actualizado y fiable. Por ello, uno de los requisitos de la nueva versión de EvALL 2.0 es la creación de un **repositorio** que almacene toda esta información de una forma sencilla y transparente para el usuario.

Conseguir llegar al mayor número de usuarios posible, a la vez que fomentar el aprendizaje de las distintas metodologías y técnicas de evaluación, es uno de los objetivos de EvALL 2.0. Para ello, se hace necesario dotar a la aplicación de una **versatilidad que permita adaptarse** a cada usuario y al contexto de uso de cada uno. En este sentido, EvALL 2.0 dispondrá de diferentes formas de ejecución para una misma evaluación (centrada en un contexto, en una sola métrica, en las métricas oficiales de una tarea, etc.), así como de diferentes tipos de informes que se adaptarán al grado de experiencia del usuario.

La nueva versión de EvALL 2.0 persigue la idea de **un proyecto a largo plazo**, aun cuando la fuente de financiación se haya agotado. Esta idea es muy importante a la hora de abordar el diseño, ya que el área de evaluación va evolucionando y van surgiendo nuevas métricas o contextos de evaluación. Por ello, es imprescindible disponer de un mecanismo sencillo y efectivo que permita incluir estos cambios. El diseño y arquitectura deben permitir el máximo grado de actualización sin necesidad de desarrollar código. Este tipo de funcionalidades suele estar ligada a *frameworks* de desarrollo como los gestores de contenido.

Para una **gestión fácil del proyecto**, se requiere que esté creado sobre una estructura de contenedores de software de virtualización como Docker. Docker es una plataforma de software que garantiza la independencia entre el entorno físico y el entorno en el que se ejecuta el software, reduciendo de este modo los problemas de compatibilidad. Además ofrece ventajas como portabilidad, velocidad de implantación y aislamiento de otros proyectos dentro del mismo servidor, optimizando al máximo el uso de los recursos disponibles.

Por último, la distribución de todo el código, especialmente la librería de evaluación PyEvALL, bajo **licencia de código abierto** es uno de los requisitos principales para llegar a perfiles de usuarios más experimentados como desarrolladores o investigadores. Así mismo, poner el código a disposición de la comunidad es, general, una garantía para la mejora del mismo mediante contribuciones y sugerencias.

2.4. Plataformas web disponibles

Se han valorado tres plataformas de desarrollo de webs, todas ellas basadas en formatos de código abierto y que cumplen, en mayor o menor medida, con las especificaciones básicas especificadas en la Sección 2.3. Describimos a continuación las tres opciones con sus ventajas e inconvenientes.

WordPress (<https://wordpress.com>) es el gestor de contenidos más popular del mundo. Según

las cifras publicadas en su web, el 40 % de las páginas web publicadas en Internet están hechas con Wordpress.

Sus principales **ventajas** son:

- Sencillez de uso. Es fácil de gestionar una vez se conocen las bases de creación y gestión de contenido, tanto en lo que se refiere a incorporación y actualización de contenidos como a labores de administración.
- Actualización constante. Las actualizaciones son prácticamente diarias, ya sea en su motor principal o de sus múltiples *plugins*.
- Gran cantidad de *plugins* y temas. Wordpress cuenta con, quizás, la mayor comunidad de desarrolladores en un proyecto de sus características. Mientras que en otros proyectos similares se puede encontrar una solución genérica para un problema determinado, en Wordpress podemos encontrar, normalmente, al menos tres soluciones más concretas para el mismo problema.
- Orientación a SEO. Quizás una de las máximas de este gestor de contenidos es su capacidad de orientación a SEO junto con los *plugins freemium* Yoast o AllInOneSEO, característica que ha ayudado enormemente a impulsar este gestor de contenidos.

Por su parte, sus principales **inconvenientes** son:

- Temas y *plugins* de pago. Pese a contar con una gran librería de temas y *plugins*, la mayor parte de éstos son en formato *freemium*/suscripción o licencia, lo que imposibilita el uso de estas herramientas para este proyecto.
- Orientación a blog. Pese a que Wordpress es un gestor de contenidos, está orientado principalmente a la creación de blogs. Cualquier funcionalidad añadida que se desee requiere la instalación de extensiones concretas que añadan dicha funcionalidad.
- Multi-idioma. Aunque de manera nativa se puede instalar Wordpress en cualquier idioma, está orientado a trabajar uno solo. Este aspecto difiere de lo que se refiere a la gestión de contenidos en diferentes idiomas. En este punto, el *plugin* más popular y que mejor trabaja este aspecto, es, sin duda, WPML (<https://wpml.org/es/>). La desventaja de este *plugin* es que es de pago.
- Seguridad. Si bien es cierto que este gestor tiene fama de inseguro, esta fama le viene dada principalmente por dos causas. La primera, porque entre los millones de páginas web desarrolladas con él, son muchas las que no están mantenidas y actualizadas convenientemente. A pesar de que su proceso de actualización es sencillo, e incluso automático en el core de las versiones 4+, es imprescindible tener actualizados tanto temas como *plugins* e incluso aquellos que se tienen desactivados. La segunda causa es que, dada la gran cantidad de páginas realizadas en este gestor, son también muchos los ojos puestos en sus vulnerabilidades, y existen múltiples *bots* encargados de rastrear y localizar versiones de Wordpress desactualizadas con configuraciones vulnerables.
- Estructura basada en *plugins* independientes. Pese a ser una de las opciones más flexibles que hay, una de las características de Wordpress que condicionan su utilización en este proyecto es que gran parte del desarrollo de la estructura básica se debería hacer por código en php, lo cual provocará que para realizar futuras mejoras o modificaciones (sencillas) sea necesario disponer de conocimientos sobre este lenguaje.

Gestor de contenidos Drupal (<https://www.drupal.org/>) es otro de los gestores de contenidos más populares hoy en día. Pese a que su cuota es muy inferior a la de Wordpress, alrededor de un 2-3 %, se pueden encontrar proyectos de alta calidad desarrollados en este gestor de contenidos. En su portal, se pueden encontrar gran cantidad de ejemplos de casos de éxito (<https://www.drupal.org/case-studies>) relevantes entre los que destacan:

- Nasa⁴

⁴<https://www.nasa.gov/>, <https://www.drupal.org/case-study/nasagov>

- Reagan Library⁵
- Stanford⁶
- Princeton⁷

Sus principales **ventajas** son:

- Entorno de desarrollo. En sus últimas versiones, se considera que Drupal está más cerca de un *framework* de desarrollo que de un gestor de contenidos. Es por ello que se recomienda para proyectos de gran alcance y que requieran una personalización determinada en cuanto a funcionamiento.
- Tipos de contenidos y vistas. Como principal ventaja para este proyecto, destacan dos módulos inherentes a este sistema, el de gestión de contenidos y el de vistas, con los que, por un lado, se puede definir casi cualquier tipo de contenido con sus campos, y por otro, crear diferentes tipos de listados de contenido adaptados a las necesidades del proyecto. Estos módulos ahorran escritura de código, a la vez que permiten añadir código para personalizar al máximo la estructura generada. Pese a que en las primeras versiones de Drupal esta funcionalidad estaba separada en dos módulos independientes, son quizás los dos módulos de mayor recorrido de la comunidad, con gran cantidad de extensiones como la posibilidad de incluir formularios embebidos.
- Estructura modular. A diferencia de Wordpress, Drupal trabaja en una estructura modular donde cada módulo puede depender o ser requisito para otros, por lo que se facilita la escalabilidad y el desarrollo de nuevas funcionalidades.
- Gestión multi-idioma integrada. Una de las grandes ventajas de Drupal para este proyecto es su gestión de contenidos multi-idioma.
- Desarrollo en línea. Otra de las características más destacables de Drupal es que está concebido para que gran parte del desarrollo e implementación de soluciones se pueda hacer sin tener que desarrollar código directamente, sino que se pueden hacer pequeñas variaciones y ampliaciones directamente desde su panel de control.
- Gran capacidad de personalización. Ya sea con los módulos inherentes como con módulos de desarrollo en bloques, Drupal permite configurar hasta el más mínimo detalle tanto su área pública como en la privada, e incluso disponer de diferentes configuraciones en función del rol o el usuario concreto.

Las posibles desventajas son:

- Una de sus principales desventajas es la poca compatibilidad de sus módulos con versiones anteriores. Es un problema que destaca sobretodo en versiones anteriores a la 7. Pese a ello, en un estudio previo del proyecto, no se ha encontrado ningún módulo de estas versiones que sea requerido, y en todo caso se puede plantear la posibilidad de un desarrollo ad-hoc que pueda suplir una necesidad concreta que no se encuentre entre el listado de módulos compatibles.
- Basado en un lenguaje de programación precompilado, lo que puede afectar en situaciones de gran carga de visitas. Este problema se puede solucionar con sistemas de *cache* internos y externos.
- Al estar programado en sistema modulares, los test unitarios tienen que ser realizados para cada módulo de manera independiente y no como un sistema integrado, lo cual puede dificultar el desarrollo continuo.
- Curva de aprendizaje pronunciada inicialmente: uno de los inconvenientes de Drupal respecto a Wordpress como gestor de contenidos, es su pronunciada curva de aprendizaje inicial.

⁵[https://www.reaganlibrary.gov/,
reagan-library-redesign](https://www.reaganlibrary.gov/,reagan-library-redesign)

⁶[https://www.gsb.stanford.edu/,
stanford-graduate-school-of-business](https://www.gsb.stanford.edu/,stanford-graduate-school-of-business)

⁷[https://spia.princeton.edu/,
princeton-university-school-of-public-and-international-affairs](https://spia.princeton.edu/,princeton-university-school-of-public-and-international-affairs)

<https://www.drupal.org/case-study/>

<https://www.drupal.org/case-study/>

<https://www.drupal.org/case-study/>

Django Framework/CMS o Flask La primera intención con Django fue la de crear un gestor de contenidos modular desarrollado en Python. Como gestor de contenidos, tiene una estructura básica bastante fiable, que puede cumplir con los requisitos básicos del proyecto.

Sus principales **ventajas** son:

- Está basado en Django y Python. Al ser un gestor de contenidos basado en Django, importa todas las ventajas de este *framework*.
- Multi-idioma. Igual que en los casos anteriores, tiene implementada una solución nativa para la gestión de lenguaje a nivel de textos básicos, pero requiere de un desarrollo concreto para esta gestión básica.
- Trabajar directamente sobre un *framework* permite alcanzar más en detalle el desarrollo a medida en funcionalidades muy concretas.
- El lenguaje de programación Python es uno de los más populares en la actualidad, y es el mismo que en el que se va a desarrollar la librería de evaluación PyEvALL, por lo que trabajar en un mismo lenguaje puede ser facilitar la integración.

Las posibles desventajas son:

- El hecho de trabajar con un *framework* obliga a desarrollar muchas funcionalidades transversales que en un gestor de contenidos ya vienen implementada por defecto, como por ejemplo la administración o la autenticación.
- Las actualizaciones en este formato vienen dadas por el *framework* en sí y por librerías añadidas, pero no por funcionalidad por lo que en un entorno de código abierto sin actualizaciones recurrentes se pueden generar graves vulnerabilidades de seguridad.
- Pese a ser multi-idioma, es necesario implementar colecciones de traducciones que en muchos casos ya se encuentran implementadas por la comunidad en los casos de Wordpress y Drupal.

2.4.1. Conclusión

Tras estas valoraciones, para este proyecto se plantea una estructura híbrida entre el gestor de contenidos Drupal, que abarca la mayor parte de las necesidades del proyecto, y la implementación de una API Rest independiente con Flask/Python que permita suplir las necesidades que el gestor no alcance.

2.5. Arquitectura de EvALL 2.0

El diseño propuesto para EvALL 2.0 sigue una arquitectura típica de sistema web en la que se pueden observar dos componentes principales: la capa de interfaz de usuario y la capa servidora. Como se puede ver en la figura 1, la capa servidora está compuesta por tres contenedores, el contenedor PHP, el contenedor de la base de datos y el contenedor API Flask.

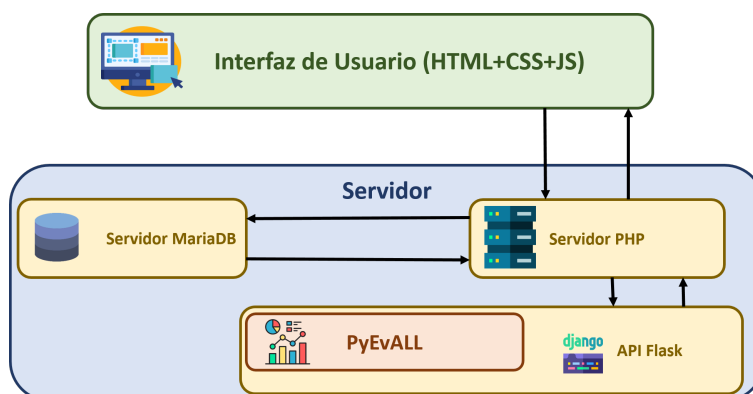


Figura 1: Arquitectura EvALL web 2.0.

Tal y como se ha comentado anteriormente, para la arquitectura de este proyecto se parte del objetivo de trabajar con contenedores aislados e implementados en Docker. De esta forma, cada parte del proyecto puede trabajar independientemente para evitar problemas de concurrencia de software, pero al mismo tiempo estos contenedores trabajan de forma unificada en un entorno de red virtualizada por el propio sistema.

Dentro de este sistema, podemos encontrar diferentes redes virtualizadas de las que destacan una red privada, a la que solo tienen acceso los contenedores a los que así se les haya configurado, y una red pública conectada con el entorno físico donde se ejecuta el sistema. Este último es el contenedor configurado como salida, por lo que se puede acceder desde Internet. De este modo, se garantiza una comunicación eficiente entre los diferentes contenedores y se maximiza la seguridad en aspectos como la base de datos y la API. Atendiendo a esto, los contenedores incluidos en la arquitectura son:

- **Contenedor Apache/PHP/Drupal:** contiene el sistema de archivos con la lógica de la programación del sistema. Es la parte central que gestiona los otros nodos del sistema. Este contenedor tiene doble conexión, una dentro del propio entorno virtualizado (privada) y otra a la red pública, dado que, por un lado, tiene que trabajar con los diferentes contenedores y, por otro, tiene que ser accesible a los usuarios.
- **Contenedor MariaDB:** contiene el SGBD MariaDB y la base de datos del proyecto. Este contenedor únicamente es accesible dentro de la red privada.
- **Contenedor API Flask:** mediante este contenedor la aplicación accede a la herramienta de evaluación PyEvALL, que es la que realmente realiza la evaluación final. Así mismo, mediante este módulo se realizan los controles necesarios para que los procesos no colapsen el servidor y los recursos del mismo. Todo este proceso se realiza mediante un conjunto de protocolos establecidos entre el contenedor PHP y este. Este contenedor únicamente es accesible dentro de la red privada.

En las siguientes secciones se detallan las funcionalidades de cada capa y la interacción entre ellas.

2.6. Capa Interfaz de usuario

Tal y como se ha comentado anteriormente, el proyecto se desarrolla en un marco de trabajo con Drupal como base. En este entorno, se ha planteado una interfaz de usuario general, distribuida en dos componentes principales, la parte pública y la parte privada. La parte pública será accesible por cualquier usuario, tanto registrado como no, mientras que la parte privada solo será accesible para usuarios registrados. Dentro de la parte pública se podrá acceder a distintas paginas de información, como pueden ser la portada o las preguntas frecuentes, o a los formularios de inicio de sesión y registro.

Como se puede ver en las figuras 2, 3 y 4, se ha diseñado una portada amigable en donde se detallan todas las funcionalidades y posibilidades que ofrece EvALL 2.0 a los usuarios, explicando su misión, sus posibilidades y sus contextos de evaluación y métricas. A través de la portada, un usuario puede navegar directamente a diferentes elementos de EvALL 2.0 como visualizar el repositorio, el FAQ o el inicio de sesión/registro. Una vez iniciada la sesión, el usuario podrá acceder a todas las opciones de evaluación, obteniendo así toda la funcionalidad de EvALL 2.0.

Además, la estructura de páginas de información se contempla como un formato híbrido entre libro de contenido y formato Wiki, donde los administradores del proyecto puedan añadir y/o editar fácilmente información que se considere relevante respecto a la evaluación de un modelo.

Una vez el usuario a accedido al sistema mediante el inicio de sesión podrá crear una evaluación y acceder al *dashboard* de evaluación, el núcleo de la parte privada de EvALL 2.0. Mediante este dashboard el usuario registrado puede realizar las distintas evaluaciones, configurar las mismas, acceder al repositorio, evaluar sus modelos contra tareas almacenadas, etc. Como se puede ver en el ejemplo de la figura 5, el *dashboard* se organiza en varias columnas, en este caso en particular, una para la configuración de la evaluación y otra para la visualización de los resultados.



Figura 2: Interfaz de la portada principal de EvALL 2.0.

2.7. Capa Servidora

Como ya se ha mencionado, la arquitectura de este proyecto se implementará mediante contenedores aislados en Docker, de tal manera que cada parte del proyecto pueda trabajar de manera independientemente para evitar problemas de concurrencia de software, aunque al mismo tiempo estos contenedores trabajan de forma unificada en un entorno de red virtualizada por el propio sistema.

Dentro de este sistema, podemos encontrar diferentes redes virtualizadas de las que destacan principalmente una privada, a la que solo tienen acceso los contenedores que así se hayan configurados, y una red pública conectada con el entorno físico donde se ejecuta el sistema. Este último dispone de conexión a Internet, por lo que será accesible por cualquier usuario. De este modo, se garantiza una comunicación eficiente entre los diferentes contenedores y se maximiza la seguridad en aspectos como la base de datos y/o la API.

Los contenedores se describen a continuación:

2.7.1. Contenedor Servidor PHP/Drupal

Dentro de este sistema, el contenedor servidor PHP/Drupal es el núcleo de interconexión entre los distintos contenedores y la capa de interfaz de usuario, donde se desarrolla la lógica de la programación del sistema. Mediante este modulo, se centralizan todas las peticiones de la parte publica, y se redirigen a los distintos contenedores privados: servidor MariaDB y API FLASK.

2.7.2. Contenedor Servidor MariaDB

Contiene el SGBD MariaDB y la base de datos del proyecto. Este contenedor únicamente es accesible dentro de la red privada.

2.7.3. Contenedor API FLASK

Este contenedor forma parte de la parte privada de EvALL 2.0. Con FLASK se ha construido un servidor API robusto capaz de gestionar múltiples peticiones. El servidor API se fundamenta en dos peticiones: una petición POST y otra GET.

En una primera petición, de tipo POST, el servidor recibirá desde la capa de interfaz de usuario el **Id de usuario, el listado de métricas seleccionadas y los ficheros para la evaluación**, a través del servidor PHP. Tras su verificación, los ficheros son reenviados a la librería PyEVALL para su procesamiento, y el



Figura 3: Interfaz de la portada principal de EvALL 2.0.



Figura 4: Interfaz de la portada principal de EvALL 2.0.

resultado es devuelto a la capa de interfaz de usuario junto con un código hexadecimal que identifica la petición actual y el **Id de usuario** que realizó la petición.

En la segunda petición, de tipo GET, la API FLASK recibe solo el código hexadecimal que identifica la petición, permitiendo con él identificar el estado de la misma. Para ello, el servidor API FLASK comprueba el estado de la petición y devuelve su status correspondiente `{running,completed}`. Si dicho status es `completed`, devolverá también los resultados de la evaluación.

2.7.4. Librería de evaluación PyEvALL

PyEvALL es la librería de evaluación de propósito general para sistemas de información de EvALL 2.0, desarrollada en el grupo de investigación de la UNED y basada en la teoría de la medida. El objetivo de esta nueva versión es convertirse en la librería de evaluación de referencia en el área de sistemas de información, y especialmente en procesamiento del lenguaje natural y recuperación de información. Dada la importancia y complejidad de esta librería, su funcionamiento detallado se describe en la sección 9.

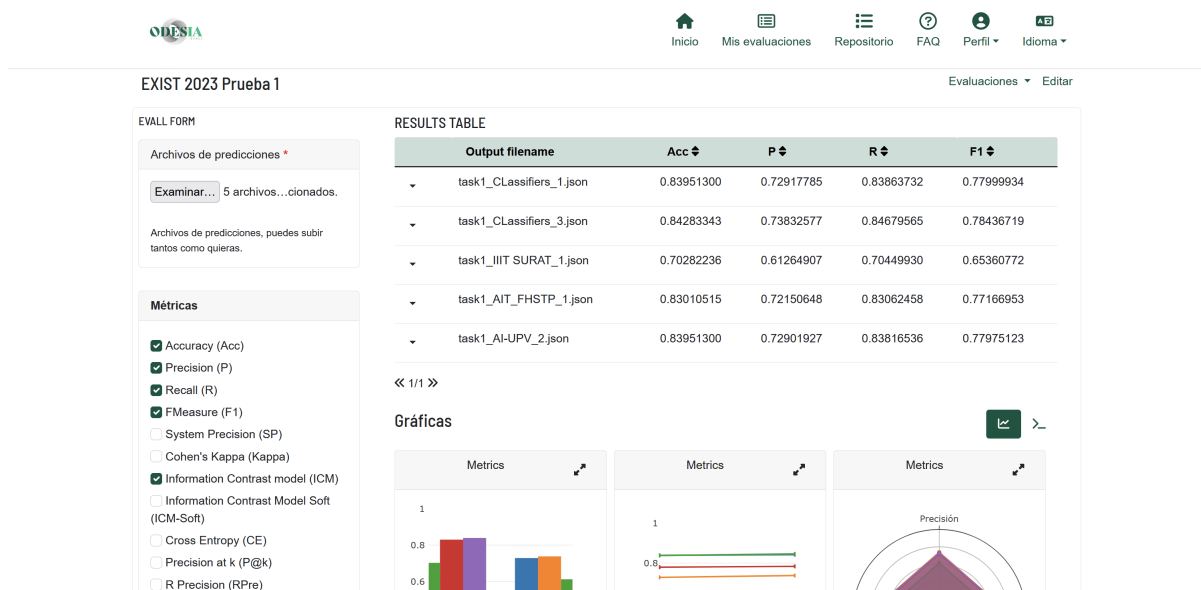


Figura 5: Interfaz del dashboard principal de EvALL 2.0.

2.8. Roles de usuario

A tenor del diseño y la arquitectura propuesta, los casos de uso, y las especificaciones de la aplicación EvALL 2.0, se identifica el siguiente conjunto inicial de roles de usuario:

- **Usuario anónimo:** en este caso, el usuario no está registrado y solo puede acceder a la parte pública de la aplicación. Cualquier usuario puede registrarse con un correo electrónico. Su solicitud debe ser aprobada por un administrador.
- **Usuario registrado:** puede acceder tanto a la parte pública como a la parte privada de la aplicación, donde podrá evaluar sus modelos o gestionar sus campañas de evaluación. Para ello, el usuario registrado debe iniciar sesión con su usuario y contraseña.
- **Usuario organizador de campaña de evaluación:** este es un rol de usuario registrado con unas características especiales sobre una tarea activa en una campaña de evaluación.
- **Usuario administrador:** el rol de administración solo está reservado para el personal de la UNED. Mediante este rol, se podrán aceptar las solicitudes de registro de los usuarios, o editar o borrar sus perfiles en casos necesarios. Así mismo, se podrá acceder y modificar el repositorio de EvALL 2.0.

3. Flujo registro de usuario

Esta sección se describe el flujo necesario para que un usuario se pueda registrar en la aplicación EvALL 2.0, y pueda así acceder a la parte privada de la misma. A continuación, se detallan las funcionalidades del flujo, los casos de uso, así como su lógica, mostrando en cada paso las pantallas implementadas para ello.

3.1. Funcionalidad

EvALL 2.0 es una aplicación orientada a usuarios registrados debido a la necesidad de controlar los procesos de evaluación en el servido para evitar un uso excesivo de recursos en las evaluaciones, así como a la necesaria supervisión que una herramienta de este tipo requiere. Por todo ello, es preciso implementar un flujo que permita a cualquier usuario anónimo registrarse en la aplicación, y poder acceder a la parte privada de la misma. Para ello, el usuario solo debe rellenar un pequeño formulario con unos datos básicos y un correo electrónico. El proceso en este prototipo se realiza en dos pasos: primero, el usuario debe cumplimentar el formulario de registro y enviarlo, y a continuación, dicha solicitud debe ser aceptada por un administrador.

Es importante tener en cuenta que los sistemas de registro e inicio de sesión son sistemas muy sensibles, en cuanto a lo que seguridad se refiere. Ante esta situación, existen dos posibilidades: (i) contar con expertos en la materia para su desarrollo; (ii) utilizar uno de los múltiples sistemas de código abierto existentes y cuya fortaleza frente ataques ha sido comprobada y evaluada en múltiples ocasiones por expertos en el área en la comunidad de código abierto. Para el desarrollo de EvALL 2.0 se ha optado por la segunda opción, utilizando el sistema de administración de usuarios que incluye por defecto el gestor de contenidos Drupal. Este sistema es de fácil configuración, y ha sido probado y usado con éxito en multitud de casos de uso, como por ejemplo, en aplicaciones de la NASA. Así mismo, la comunidad de código abierto detrás del desarrollo de Drupal proporciona actualizaciones frecuentes para solucionar posibles fallos de seguridad.

Para la correcta realización de este proceso, se hace necesario un identificador clave para cada usuario, que en este caso es el correo electrónico. Como es de esperar, a lo largo del proceso el sistema comprueba que no haya usuarios con el mismo identificador solicitado, ni el mismo correo electrónico, informando en cada caso del error pertinente.

3.2. Casos de uso

Para la realización de este flujo se han tenido en cuenta los siguiente roles de usuario y casos de uso asociados:

3.2.1. Usuario - Anónimo

La plataforma está restringida a usuarios registrados, por lo que los usuarios anónimos solo pueden acceder a la parte pública de la aplicación. Pese a ello, se contempla un entorno público en el que usuarios anónimos pueden visualizar y realizar las siguientes acciones relacionadas con este flujo:

- **Visualizar los accesos al flujo de registro:** el usuario anónimo podrá visualizar la portada de la aplicación y visualizar el acceso a los diferentes elementos de la misma, entre ellos el acceso al registro de usuario.
- **Acceder al formulario de inicio de sesión y registro:** la aplicación mostrará la pantalla para iniciar sesión en la que se mostrarán las opciones de *Inicio de sesión*, *Crear nueva cuenta* y *Reinicializar su contraseña*.
- **Cumplimentar el formulario de registro:** el usuario podrá cumplimentar y enviar el formulario de registro proporcionando unos datos personales básicos y un correo electrónico.

3.2.2. Usuario - Administrador

En este flujo se hace precisa la intervención de un administrador de la herramienta para la aceptación de la solicitud de registro:

- **Aceptar las solicitudes de registros pendientes:** el usuario administrador podrá aceptar o rechazar las solicitudes de registro pendientes mediante un panel de administración. Una vez aceptada la solicitud de registro, automáticamente se enviará un correo al usuario con la información necesaria para formalizar el registro en la aplicación.

3.2.3. Usuario - Registrado

Los usuarios aceptados por el administrador pasarán automáticamente a ser usuarios registrados, pero antes de poder acceder a la parte privada deberán validar la cuenta:

- **Validar la cuenta de usuario:** el usuario ya registrado deberá validar la cuenta mediante el enlace enviado a su correo. En este proceso de validación, el usuario deberá, a su vez, modificar la contraseña establecida automáticamente en el proceso de registro.

3.3. Lógica del flujo

La lógica de este flujo solo es válida para usuarios que no han iniciado la sesión, por lo que solo se tiene en cuenta para este flujo el rol de usuario anónimo. A continuación, se describe, paso a paso, la lógica del flujo para realizar el registro de un usuario nuevo.

1. En primer lugar, el usuario debe acceder al formulario de inicio de sesión y registro. Para ello, tal y como se puede ver en la figura 13, el usuario debe pulsar sobre el icono de *Login/Registro* del menú superior derecho de la pantalla.

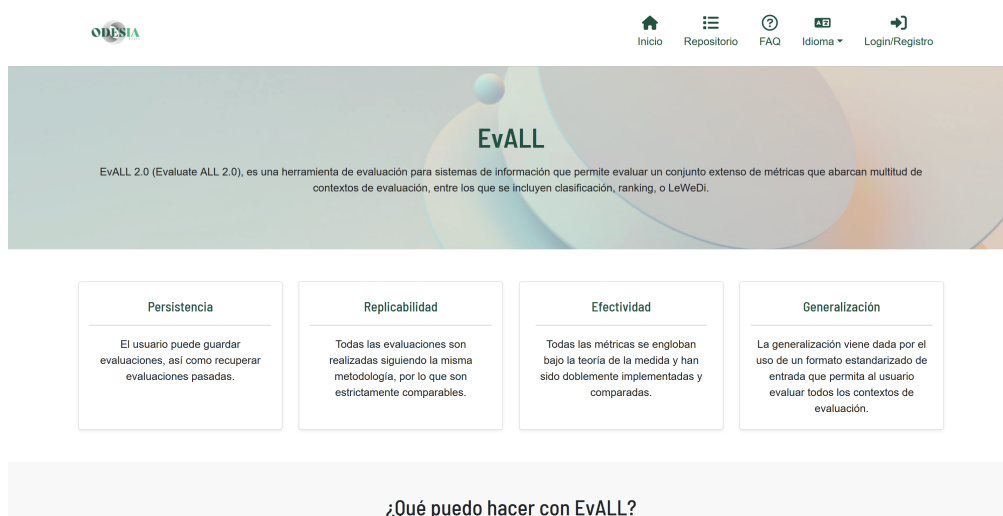



Figura 6: Interfaz de la portada principal de EvALL 2.0.

2. Una vez que el usuario ha accedido a este menú, el formulario presenta diferentes opciones de registro e inicio de sesión. Para realizar el registro, el usuario debe elegir la opción de *Crear nueva cuenta* en la siguiente pantalla 7.
3. En el formulario *Crear nueva cuenta*, tal y como se puede ver en la imagen 8, el usuario debe rellenar los campos obligatorios de correo electrónico y nombre de usuario. Una vez introducidos todos los datos, el usuario debe pulsar sobre el botón *Crear cuenta nueva*. Así mismo, la creación del perfil supone la aceptación de todos los términos de uso de la aplicación.
4. Tras el envío del formulario, la solicitud del usuario será registrada y marcada como pendiente de aceptación. Para formalizar el registro, un administrador debe aceptar la solicitud mediante el panel de administración de usuarios, lo que enviará al usuario un correo electrónico con un enlace de un solo uso.



Inicio
Repositorio
FAQ
Idioma
Login/Registro

INICIAR SESIÓN

Iniciar sesión
Crear nueva cuenta
Reinicializar su contraseña


Nombre de usuario *

Escriba su nombre de usuario en EvALL.


Contraseña *

Escriba la contraseña asignada a su nombre de usuario.


Iniciar sesión




Financiado por la Unión Europea
NextGenerationEU




GOBIERNO DE ESPAÑA



MINISTERIO DE TRANSFORMACIÓN DIGITAL








Figura 7: Interfaz de registro de usuario de EvALL 2.0.

5. Por último, el usuario debe validar, mediante la información enviada en el correo electrónico, el registro en la aplicación. Así mismo, deberá generar una contraseña propia, la cual será la que deba utilizar en el inicio de sesión a la aplicación.



Inicio
Repositorio
FAQ
Idioma
Login/Registro

CREAR NUEVA CUENTA

Iniciar sesión
Crear nueva cuenta
Reinicializar su contraseña

Dirección de correo electrónico *

The email address is not made public. It will only be used if you need to be contacted about your account or for opted-in notifications.

Nombre de usuario *

Varios caracteres están permitidos, incluyendo los espacios, puntos (.), guiones (-), comillas (") y el signo @.

País

Afilación

Crear nueva cuenta

Figura 8: Formulario de registro de usuario de EvALL 2.0.

4. Flujo inicio de sesión

Esta sección describe el flujo necesario para que un usuario registrado inicie sesión en la aplicación de EvALL 2.0 y pueda, así, acceder a la parte privada de la misma. A continuación se detallan las funcionalidades del flujo, los casos de uso, y su lógica, mostrando en cada paso las pantallas implementadas para ello.

4.1. Funcionalidad

EvALL 2.0 es una aplicación orientada a usuarios registrados, por lo que es necesario proporcionar al usuario ya registrado un flujo que le permita iniciar y acceder a la parte privada. Para ello, el usuario solo debe rellenar un pequeño formulario con los datos de acceso, nombre de usuario y contraseña, y enviar la información para su comprobación. Como es lógico, todo sistema de credenciales de inicio de sesión necesita de un sistema de verificación tanto para el nombre de usuario como para la contraseña, comunicando al usuario cualquier tipo de error, tal y como se puede ver la siguiente figura 9.

The screenshot shows the 'INICIAR SESIÓN' (Login) page of the EvALL 2.0 application. At the top, there is a navigation bar with links: Inicio, Repositorio, FAQ, Idioma, and Login/Registro. Below the navigation bar, the title 'INICIAR SESIÓN' is centered. Underneath, there are three tabs: 'Iniciar sesión' (selected), 'Crear nueva cuenta', and 'Reiniciar su contraseña'. The 'Iniciar sesión' tab contains a form with two input fields: 'Nombre de usuario' and 'Contraseña'. The 'Nombre de usuario' field contains the text 'asdf'. Below this field, a red error message is displayed: 'Usuario o contraseña no reconocidos. ¿Olvidaste tu contraseña?'. Below the error message, there is a link that says 'Escribe el nombre de usuario en EvALL.' and a 'Contraseña' field. Below the 'Contraseña' field, there is a link that says 'Escribe la contraseña asignada a su nombre de usuario.' and a green 'Iniciar sesión' button.

Figura 9: Error en el inicio de sesión de EvALL 2.0.

Es importante mencionar que todas las contraseñas están cifradas en el sistema, por lo que son inaccesibles incluso para los administradores. Por ello, el sistema debe de proporcionar a los usuarios un mecanismo de recuperación de contraseña o regeneración. En el caso de EvALL 2.0, el usuario puede regenerar la contraseña mediante un proceso en dos pasos. En un primer paso, el usuario indica al sistema el correo electrónico o usuario, lo que provoca que el sistema compruebe si dicho usuario existe. En caso afirmativo, el sistema manda un correo electrónico con un token de seguridad. El usuario, en un segundo paso, deberá acceder con ese token a la aplicación e introducir una nueva contraseña, que sustituirá a la anterior.

4.2. Casos de uso

Para la realización de este flujo se han tenido en cuenta los siguiente roles de usuario y casos de uso asociados:

4.2.1. Usuario - Anónimo

En el caso de inicio de sesión, la plataforma solo contempla opciones para usuarios anónimos. Para ello, se proporciona el entorno público en el que usuarios anónimos pueden visualizar y realizar las siguientes acciones relacionadas con este flujo:

- **Visualizar los accesos al flujo de inicio de sesión:** el usuario anónimo podrá visualizar la portada de la aplicación y el acceso a los diferentes elementos de la misma, entre ellos, el acceso al inicio de sesión.
- **Acceder al formulario de inicio de sesión y registro:** la aplicación mostrará la pantalla para iniciar sesión en la que se mostrarán las opciones de *Inicio de sesión*, *Crear nueva cuenta* y *Reiniciar su contraseña*.

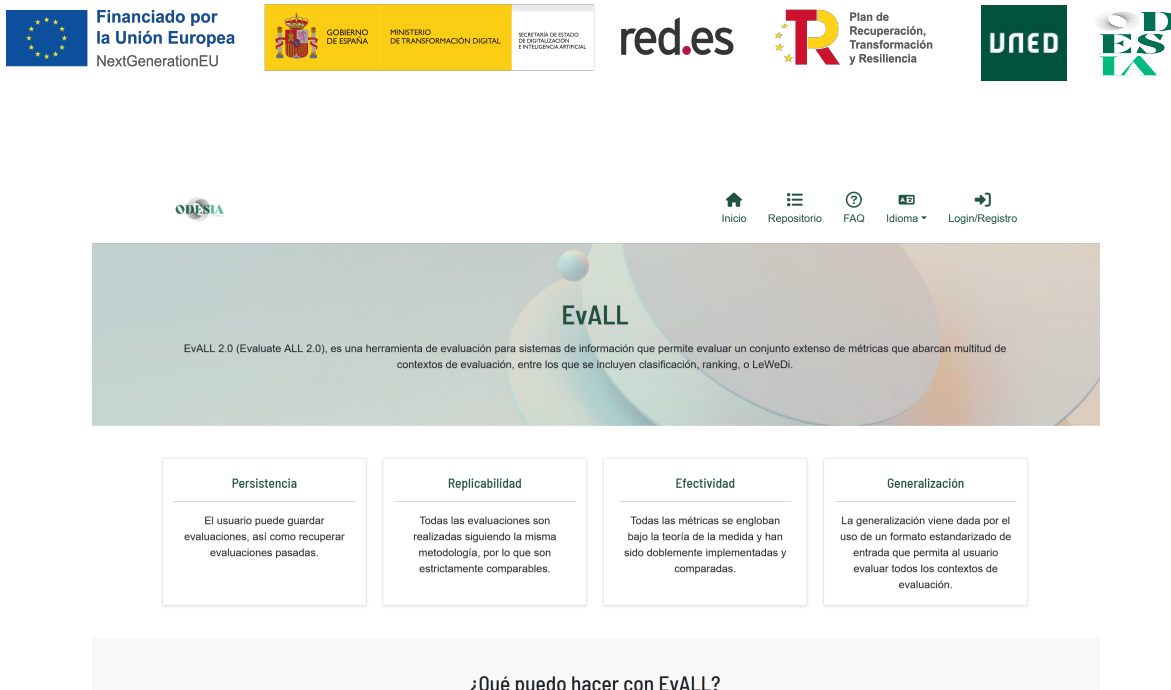


Figura 10: Interfaz de la portada principal de EvALL 2.0.

- **Cumplimentar del formulario de inicio de sesión:** el usuario podrá cumplimentar y enviar el formulario de inicio de sesión. El sistema responderá iniciando sesión, en el caso de usuarios previamente registrados, o mostrando un error en el caso de usuarios no registrados.

4.3. Lógica del flujo

La lógica de este flujo solo es válida para usuarios que no han iniciado la sesión, por lo que solo se tiene en cuenta para este flujo el rol de usuario anónimo. A continuación, se describe paso a paso la lógica del flujo para realizar el inicio de sesión en la aplicación.

1. En primer lugar, el usuario debe acceder al formulario de inicio de sesión y registro. Para ello, tal y como se puede ver en la figura 10, el usuario debe pulsar sobre el icono de *Login/Registro* del menú superior derecho de la pantalla.
2. Una vez que el usuario ha accedido a este menú, el formulario presenta diferentes opciones de registro e inicio de sesión. Para iniciar sesión, el usuario debe elegir la opción de *Iniciar sesión* (ver figura 11).

Figura 11: Interfaz de inicio de sesión de usuario de EvALL 2.0.

3. En el formulario *Iniciar sesión* de la figura 11, el usuario debe rellenar los campos con su nombre de usuario y contraseña, y pulsar el botón *Iniciar sesión*.

4. Es importante mencionar que, en el caso de que un usuario registrado haya olvidado la contraseña, se ofrece la opción de regenerar la misma mediante el formulario *Reinicializar su contraseña*, tal y como se puede ver en la figura 12.



REINICIALIZAR SU CONTRASEÑA

[Iniciar sesión](#) [Crear nueva cuenta](#) [Reinicializar su contraseña](#)

Nombre de usuario o correo electrónico *

Las instrucciones para restablecer la clave se enviarán a la dirección de correo electrónico con la que se registró como usuario.

[Enviar](#)

Figura 12: Formulario para regenerar la contraseña de usuario de EvALL 2.0.

5. Flujo consulta del repositorio EvALL 2.0

En esta sección se describe el flujo a través del cual un usuario cualquiera, ya haya iniciado la sesión o no, puede visualizar los *gold standard* del repositorio de EvALL 2.0 y sus tareas asociadas. Así mismo, accediendo al enlace del título de cada uno de los *gold standard*, se puede visualizar su *leaderboard*. Este *leaderboard* está compuesto por todos los resultados asociados a esa tarea y publicados por los usuarios de EvALL, así como los resultados transferidos por el equipo de investigadores de la UNED desde la aplicación Leaderboard (<http://leaderboard.odesia.uned.es/>) y/o Porta ODESIA (<http://portal.odesia.uned.es/>). A continuación, se detallan las funcionalidades del flujo, los casos de uso y su lógica, mostrando en cada paso las pantallas implementadas para ello.

5.1. Funcionalidad

La funcionalidad principal de este flujo consiste en permitir a los usuarios visualizar el contenido del repositorio EvALL 2.0, mostrando los *gold standard* incluidos en el mismo, así como su información asociada: tarea, dataset, temática, idioma, etc. Toda esta información asociada al *gold standard* es extraída de la infraestructura común que conforman las tres aplicaciones del ODESIA: el portal, el *leaderboard* y EvALL 2.0. Así mismo, mediante el repositorio de EvALL 2.0 se puede acceder al *leaderboard* individual de cada tarea donde el usuario puede visualizar los resultados publicados por otros usuarios, o los procedentes de las aplicaciones Portal y Leaderboard ODESIA y transferidos por el equipo UNED. Además, la página de cada *leaderboard* incluye información adicional como descripción de la tarea, enlace a la publicación, etc. Mediante esta característica, EvALL 2.0 permite a cualquier usuario acceder al estado del arte de un amplio conjunto de tareas en un mismo sitio y de una forma fácil y sencilla.

5.2. Casos de uso

Para la realización de este flujo, se han tenido en cuenta los siguiente roles de usuario y casos de uso asociados:

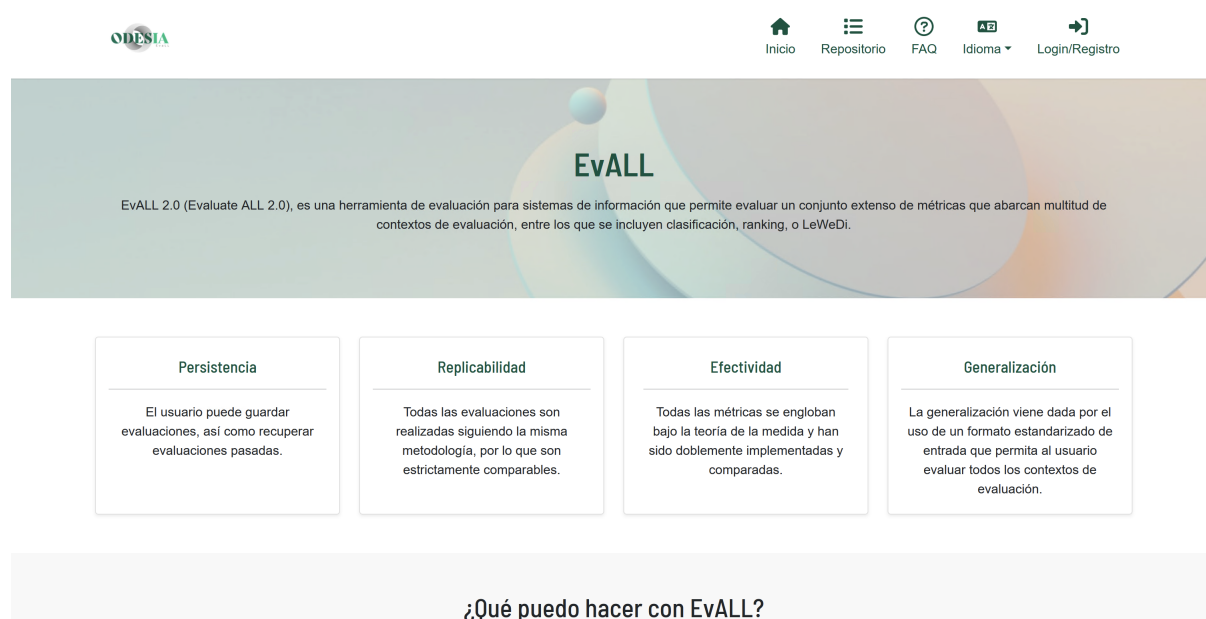


Figura 13: Interfaz de la portada principal de EvALL 2.0.

5.2.1. Usuario - Anónimo y Registrado

Este flujo está pensado para que cualquier usuario, tanto anónimo como registrado, pueda visualizar el contenido del repositorio EvALL 2.0, es por ello que cualquiera de los roles de usuario pueden acceder a:

- **Visualizar el acceso al repositorio EvALL 2.0:** el usuario anónimo o registrado podrá visualizar el acceso al repositorio EvALL mediante el menú superior.
- **Acceder a la página del repositorio EvALL 2.0 y visualizar los *gold standards* disponibles:** la aplicación mostrará la pantalla de visualización del repositorio EvALL 2.0, con estética similar a todas las desarrolladas en el proyecto ODESIA, y donde el usuario podrá visualizar todos los *gold standards* incluidos, así como su meta-información.
- **Acceder a los datos y al *leaderboard* de cada *gold standard*:** el usuario podrá acceder, pulsando en el título de un *gold standard*, a la página individual de información de ese *gold standard* donde, además, podrá visualizar su *leaderboard*.

5.3. Lógica del flujo

A continuación, se describe, paso a paso, la lógica del flujo para la consulta del repositorio EvALL 2.0, así como la página de información completa de un *gold standard* con su *leaderboard* asociado.

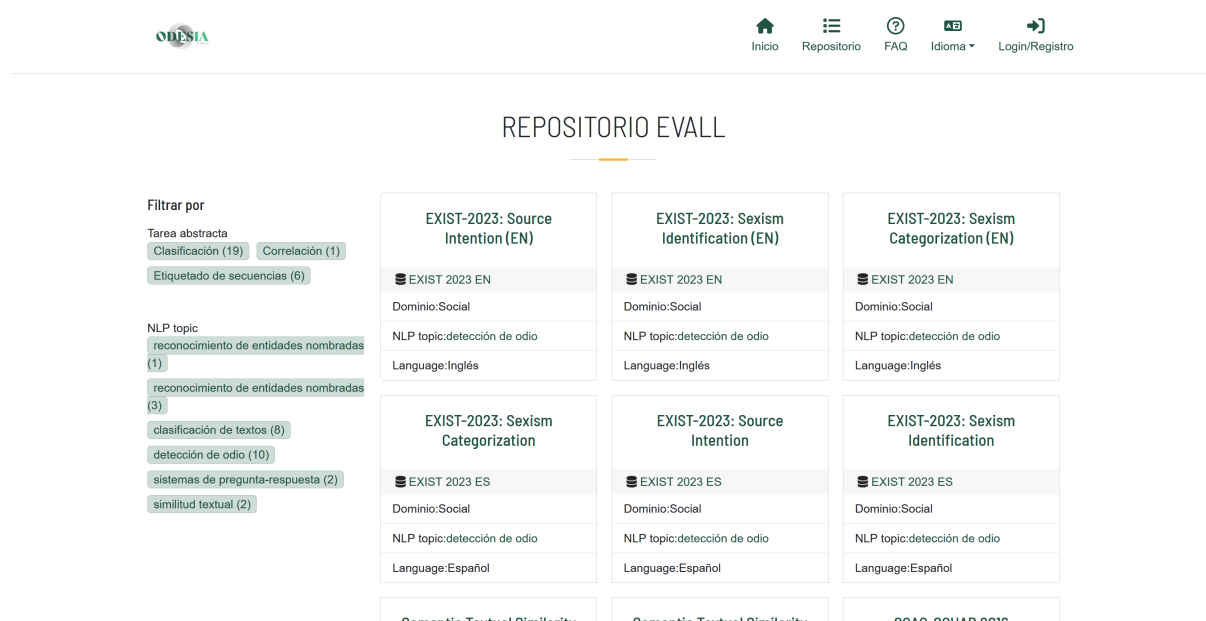


Figura 14: Interfaz del repositorio de EvALL 2.0.

1. El acceso al repositorio EvALL se encuentra disponible en el menú principal mediante el icono *Repositorio*, accesible desde la portada de la página web, tal y como se puede ver en la figura 13.
2. Una vez el usuario haya pulsado en el icono, se visualizará la pantalla del repositorio EvALL 2.0, tal y como se puede ver en la figura 14. La nueva interfaz sigue una estética similar a la utilizada en todo el proyecto ODESIA, donde cada cuadrado representa un *gold standard* incluido en el repositorio así como su meta-información.
3. Así mismo, el usuario puede aplicar filtros para acotar sus opciones de búsqueda y encontrar de una forma más sencilla y eficaz el *gold standard* deseado, visualizar su meta-información o acceder a su *leaderboard*. La figura 15 muestra los filtros aplicados (*clasificación* y *clasificación de textos*). Cabe destacar que la conjunción de varios filtros genera un comportamiento de tipo *OR* lógico.
4. Finalmente, el usuario puede acceder al *leaderboard* y a la información extendida de cada *gold standard* pulsando sobre el título de cada *gold standard*, tal y como se puede ver en la figura 16.

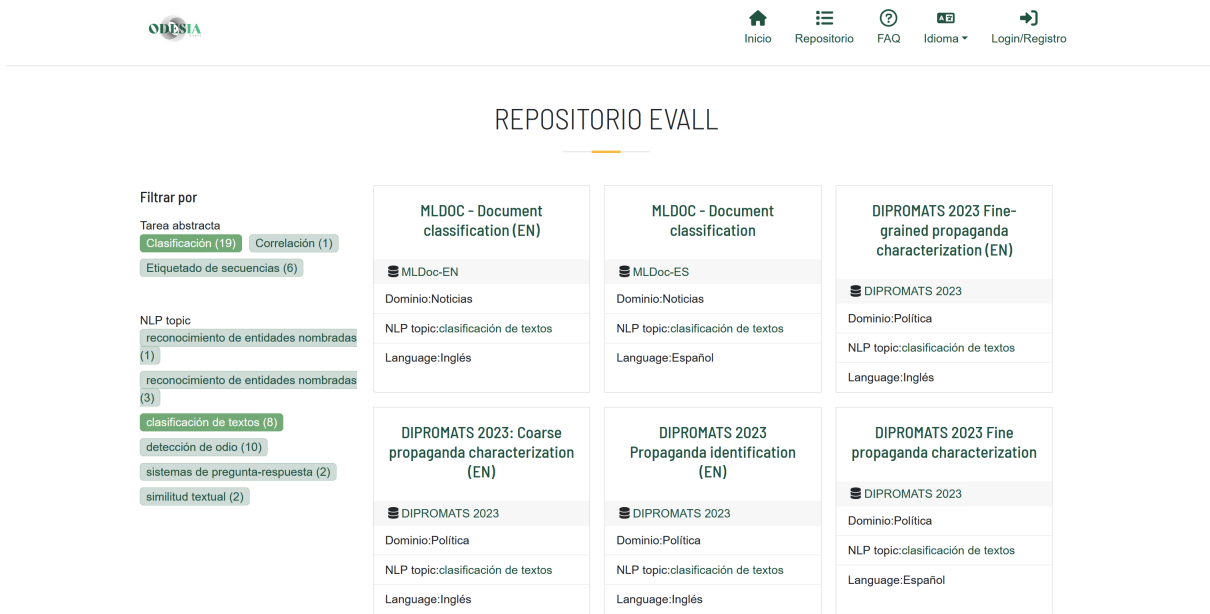


Figura 15: Interfaz del repositorio de EvALL 2.0 con filtros.

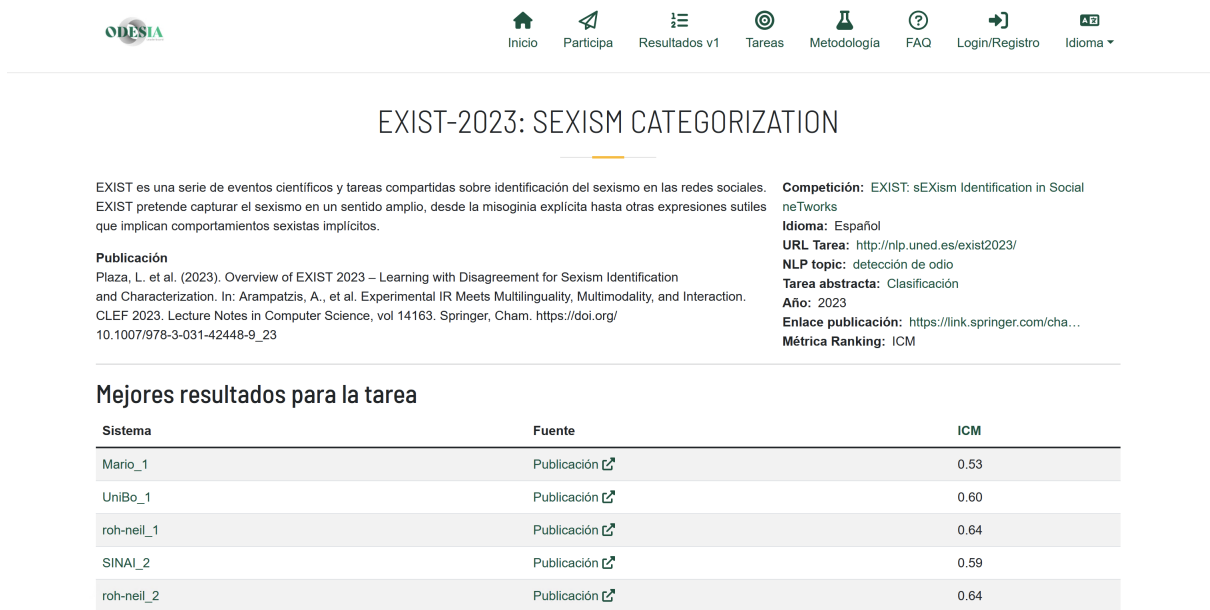


Figura 16: Interfaz del repositorio de EvALL 2.0 mostrando el leaderboard de un *gold standard*.

6. Flujo gestión de evaluaciones

En esta sección, se describe el flujo a través del cual un usuario registrado puede visualizar y gestionar sus evaluaciones, así como crear nuevas, ya sea mediante un *gold standard* proporcionado por el usuario o mediante uno disponible en el repositorio EvALL 2.0. Además, mediante esta interfaz el usuario registrado puede visualizar estadísticas de su actividad en la aplicación EvALL 2.0, como pueden ser las evaluaciones creadas o los resultados obtenidos. A continuación, se detallan las funcionalidades del flujo, los casos de uso y su lógica, mostrando en cada paso las pantallas implementadas para ello.

6.1. Funcionalidad

La funcionalidad principal de este flujo consiste en permitir a los usuarios registrados acceder a sus evaluaciones realizadas, eliminarlas o crear nuevas. La página de gestión de evaluaciones, o *Mis evaluaciones*, es el centro neurálgico de la aplicación EvALL 2.0, que permite el acceso al *dashboard* de evaluación. Como ya se ha expuesto en varias ocasiones en este informe, en EvALL 2.0 se entienden las evaluaciones como un proceso que va evolucionando con el tiempo, para lo cual toda actividad debe ser guardada automáticamente y ser accesible en cualquier momento. Este es precisamente el funcionamiento de este flujo.

Mediante la página *Mis evaluaciones*, un usuario podrá acceder a sus evaluaciones pasadas para visualizar sus resultados, evaluar nuevos sistemas y compararlos con los anteriores, borrar evaluaciones almacenadas en la base de datos, o crear nuevas evaluaciones. Finalmente, el usuario podrá visualizar ciertas estadísticas sobre su actividad en EvALL 2.0, como pueden ser el número de evaluaciones creadas, el resumen de los resultados obtenidos o los resultados publicados en el repositorio.

6.2. Casos de uso

Para la realización de este flujo, se han tenido en cuenta los siguientes roles de usuario y casos de uso asociados:

6.2.1. Usuario - Registrado

En el caso de gestión de evaluaciones, la plataforma solo contempla opciones para usuarios registrados. Por tanto, el usuario registrado tiene pleno acceso a este flujo que consta de las siguientes acciones:

- **Visualizar el acceso a Mis Evaluaciones:** el usuario registrado podrá visualizar la portada de la aplicación y el acceso a la opción *Mis evaluaciones* en el menú superior.
- **Acceder a la interfaz de gestión de evaluaciones:** el portal mostrará al usuario la interfaz mediante la cual podrá gestionar sus evaluaciones almacenadas automáticamente en la base de datos.
- **Eliminar una evaluación almacenada:** el usuario, mediante el botón con icono de papelera, podrá eliminar una evaluación almacenada automáticamente en la base de datos.
- **Acceder al dashboard de una evaluación:** pulsando en el título de una evaluación, el usuario podrá acceder al *dashboard* de dicha evaluación donde podrá visualizar los resultados obtenidos previamente, subir nuevos archivos de predicciones y evaluarlos, analizar las gráficas, etc.
- **Crear una nueva evaluación:** pulsando sobre el botón *Nueva Evaluación*, el usuario accederá a la interfaz para crear una nueva evaluación, ya sea subiendo su propio *gold standard* o utilizando uno ya subido previamente en el repositorio.

6.3. Lógica del flujo

La lógica de este flujo solo es accesible para usuarios registrados, por lo que todo usuario anónimo que intente acceder a la interfaz de gestión de evaluaciones sin estar registrado deberá iniciar sesión previamente (registrándose si fuera necesario). A continuación, se describe la lógica del flujo paso a paso, asumiendo que el usuario ya está registrado y ha iniciado la sesión, para visualizar y gestionar las evaluaciones del usuario.



Figura 17: Interfaz de la portada principal de EvALL 2.0 con usuario registrado.

1. En primer lugar, el usuario registrado debe acceder a la pantalla de gestión de sus evaluaciones. Para ello, tal y como se puede ver en la figura 17, el usuario debe pulsar en el botón *Mis evaluaciones*, en el menú superior de la aplicación.

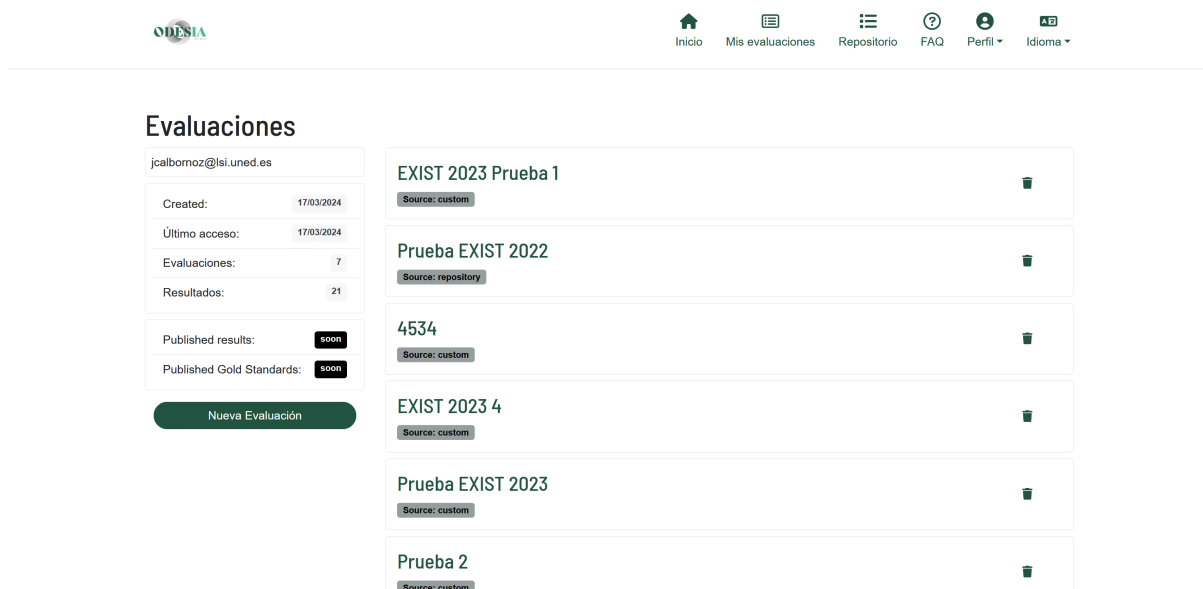


Figura 18: Interfaz de EvALL 2.0 para la gestión de evaluaciones almacenadas en base de datos.

2. Una vez el usuario haya pulsado el botón de *Mis evaluaciones*, la aplicación mostrará la pantalla de gestión de evaluaciones, tal y como se puede ver en la figura 18. La pantalla se ha dividido en dos columnas. Por un lado, el panel izquierdo, con una breve información sobre el perfil del usuario junto con un resumen de las estadísticas de la actividad del usuario en la aplicación: evaluaciones creadas, resultados obtenidos, fecha de último acceso, etc. Y por otro lado, una lista con todas las evaluaciones almacenadas automáticamente en la base de datos para ese usuario así como la información del tipo de evaluación: *gold standard* desde repositorio o proporcionado por el usuario. Esta lista da acceso a las tres acciones principales de esta pantalla:

- a) **Acceder al *dashboard* de una evaluación:** para realizar esta acción, el usuario simplemente debe pulsar sobre el título de la evaluación y EvALL 2.0 mostrará la interfaz de *dashboard* con todos los resultados almacenados para esa evaluación, así como las gráficas generadas para la misma.
- b) **Eliminar una evaluación existente:** esta opción es fácilmente accesible desde la lista de evaluaciones, a través del icono de papelera disponible en cada elemento de la lista. Una vez pulsado ese icono, el sistema pedirá confirmación para su eliminación. Si el usuario confirma, esa evaluación y todos sus datos asociados serán eliminados de la base de datos.
- c) **Crear una evaluación nueva:** el botón *Nueva Evaluación* permite al usuario acceder al formulario de creación de una nueva evaluación, tal y como se observa en la figura 19. El formulario permite introducir un nombre para la evaluación, que se aconseja sirva de recordatorio para el usuario de que tipo de evaluación realizó, así como diferentes campos que variarán dependiendo de la evaluación que seleccione: *Evaluar utilizando un gold standard desde repositorio* o *Evaluar proporcionando tu propio gold standard*.

Nueva evaluación

Descripción:

Título *

Configuración:

"Gold standard" for an NLP task is a data set of natural language texts annotated by humans for correct solutions of that particular task

Tipo de evaluación *

Evaluar utilizando un gold standard desde repositorio

Seleccionar Gold Standard de repositorio *

Escoge uno de los datasets de nuestro repositorio

Guardar

Figura 19: Interfaz de EvALL 2.0 para crear una nueva evaluación.

7. Flujo de evaluación proporcionando el *gold standard*

Esta sección describe el flujo de evaluación de EvALL 2.0 que proporciona soporte a investigadores o desarrolladores para la evaluación de sus sistemas de información. En esta modalidad de evaluación, es el usuario el que proporciona tanto los archivos con las predicciones de los modelos como el *gold standard*. A continuación, se detallan en profundidad las funcionalidades del flujo, los casos de uso y su lógica, mostrando en cada paso las pantallas implementadas para ello.

7.1. Funcionalidad

Como ya se ha expuesto, el objetivo de este flujo de evaluación es dotar a los usuarios de EvALL 2.0 de una herramienta que permita acceder a un conjunto amplio de contextos de evaluación y métricas con las que evaluar sus modelos de predicción frente a un *gold standard* proporcionado por el usuario. Así mismo, se plantea como requisito que todo el proceso sea amigable y transparente, independientemente del grado de experiencia del usuario. Por ello, en este flujo se propone una interfaz mediante la cual un usuario puede realizar una evaluación en cuatro simples pasos: (i) configurar la evaluación (subir el archivo *gold standard* y determinar formatos); (ii) subir uno, o varios, archivos con predicciones; (iii) seleccionar las métricas deseadas; (iv) pulsar el botón *Evaluar*.

Para ello, se ha seleccionado el formato de panel, o *dashboard* en inglés, que permite visualizar toda la información necesaria para realizar la evaluación en una sola pantalla. Mediante este panel, el usuario podrá, por un lado, gestionar la configuración de la evaluación (subir archivos, seleccionar métricas incluidas en la evaluación, etc.), y por otro, visualizar los resultados de las distintas evaluaciones.

Es importante mencionar que en EvALL 2.0 el proceso de evaluación está centrado en el concepto de *gold standard*, sobre el que se realizan los cálculos de las distintas métricas con los distintos archivos de predicciones. Es, por tanto, el par *gold standard* - predicción el núcleo de la interfaz, sobre el que se ejecutaran las métricas seleccionadas y bajo los parámetros configurados. Para facilitar el funcionamiento de cada evaluación sobre un mismo *gold standard*, la aplicación permite al usuario realizar evaluaciones tanto 1-1 como 1-n (un *gold standard* - un archivo de predicciones y un *gold standard* - n archivos de predicciones). Los resultados obtenidos se muestran en formato tabla donde el usuario puede analizar en detalle cada uno de ellos, ordenándolos a su gusto mediante las distintas opciones proporcionadas. Además, EvALL 2.0 permite cierto control de la gestión de estos resultados, como es la eliminación de filas de resultados del sistema.

Mediante esta pantalla, el usuario podrá generar los informes PyEvALL de cada ejecución para el par archivos de predicciones - *gold standard*, pudiendo visualizar los posibles errores en las pre-condiciones, en los formatos, etc., así como los resultados desagregados por *test cases* o clases de predicciones (en contextos de evaluación como clasificación y métricas que lo permitan como *precision* o *recall*). Así mismo, el usuario podrá acceder a la consola PyEvALL, donde se pueden analizar, a más alto nivel, los posibles errores ocurridos durante la ejecución de cada evaluación.

Finalmente, mediante esta pantalla el usuario puede analizar las gráficas propuestas por la aplicación donde podrá jugar con los distintos archivos de predicciones o métricas visualizadas. Estas gráficas permiten también realizar fácilmente capturas de pantalla, que pueden ser posteriormente utilizadas en artículos científicos o memorias de proyectos.

7.2. Casos de uso

Para la realización de este flujo, la plataforma solo contempla opciones para usuarios registrados.

7.2.1. Usuario - Registrado

El rol de usuario registrado tiene pleno acceso al flujo de evaluación proporcionando el *gold standard*:

- **Visualizar los accesos al flujo de evaluación proporcionando un *gold standard*:** el usuario registrado podrá visualizar estas evaluaciones en la interfaz de gestión de evaluaciones accesible mediante el botón *Nueva Evaluación*.

- **Acceder al *dashboard* de evaluación:** la aplicación mostrará el *dashboard* y cargará automáticamente cualquier resultado previamente realizado y asociado al *gold standard* de dicha evaluación.
- **Realizar una evaluación:** en este caso, el usuario puede evaluar un sistema comparando el *gold standard* y uno o varios archivos de predicción generados para este mediante el formulario que se encuentra en el *dashboard* de la plataforma. Para ello, el usuario deberá generar una nueva evaluación seleccionando la opción *Evaluar proporcionando tu propio gold standard*. Una vez generada la evaluación, y ya en el *dashboard*, el usuario deberá subir los archivos de predicciones, seleccionar las métricas deseadas y pulsar sobre el botón *Evaluar*. Es importante mencionar que todas las evaluaciones realizadas son guardadas automáticamente.

Figura 20: Formulario nueva evaluación de EvALL 2.0.

- **Comparar resultados de varios modelos:** los resultados obtenidos de diferentes evaluaciones se muestran en formato tabla, donde cada fila es un archivo de predicción y cada columna una métrica de evaluación. En caso de ocurrir algún error en la ejecución el sistema, se mostrará el carácter `.en` en la tabla, y el usuario podrá analizar en detalle el error mediante la consola PyEvALL o el informe PyEvALL asociado a esa ejecución. Además, el usuario podrá ordenar los resultados por métrica, tanto en orden ascendente como descendente, mediante los botones con forma de triángulo incluidos en cada columna.
- **Eliminar un resultado de la tabla:** el usuario podrá, mediante el botón *Delete* incluido en cada fila de la tabla, eliminar esa fila de resultados de la evaluación.
- **Visualizar el informe PyEvALL para ese archivo de predicción:** el usuario podrá, mediante el botón informe incluido en cada fila de la tabla, visualizar el informe detallado de PyEvALL, pudiendo identificar los distintos errores ocurridos, si los hubiera, así como los resultados desagregados por *test cases* o clases de predicción.
- **Visualizar la consola PyEvALL:** mediante esta consola, el usuario podrá visualizar de una forma rápida los posibles errores ocurridos en toda la evaluación.
- **Visualización gráfica de los resultados:** EvALL 2.0 proporciona un conjunto de gráficas generadas automáticamente en base a los resultados obtenidos en la evaluación. Estas gráficas permiten al usuario comparar los resultados de las distintas métricas con los distintos archivos de predicciones, pudiendo incluso tomar capturas de pantalla y guardarlas para su posterior uso.

7.3. Lógica del flujo

La lógica de este flujo solo es accesible para usuarios registrados. A continuación se describe paso a paso, asumiendo que el usuario ya está registrado, y desea realizar una evaluación proporcionando el *gold standard*.

1. En primer lugar, el usuario registrado debe acceder a la interfaz de gestión de evaluaciones y pulsar el botón *Nueva Evaluación*.
2. Una vez en el formulario de creación de evaluaciones, el usuario debe seleccionar la opción *Evaluar proporcionando tu propio gold standard*, tal y como se aprecia en la figura 20.
3. En el formulario de creación de evaluación subiendo el *gold standard*, el usuario debe introducir los datos de la evaluación, subir el *gold standard* e indicar el formato del mismo. Además, EvALL 2.0 proporciona la opción de subir un archivo de jerarquía en caso de que la evaluación lo precise, tal y como se ve en la figura 21. Finalmente, el usuario debe pulsar sobre el botón *Guardar*.

Figura 21: Formulario nueva evaluación de EvALL 2.0 cumplimentado.

4. Una vez que el usuario ha pulsado sobre el botón *Guardar*, la aplicación muestra el *dashboard* de evaluación sin resultados, tal y como se observa en la figura 22. La pantalla se ha dividido en dos columnas: por un lado, la configuración de la evaluación, y por otro, la visualización de los resultados. En la primera, el usuario puede encontrar tanto el formulario de evaluación, donde seleccionar los archivos de predicciones, como la selección de las métricas. En la segunda, se muestra una tabla con los resultados obtenidos en la evaluación, así como las gráficas generadas automáticamente y el acceso a la consola PyEvALL.
5. El siguiente paso a realizar será subir el archivo, o archivos, con las predicciones de los modelos. En este caso, EvALL 2.0 sí permite subir varios archivos de predicciones de distintos modelos a la vez y que serán evaluados contra el mismo *gold standard*.
6. El último paso de la configuración de la evaluación es la selección de las métricas. En el panel de configuración, el usuario puede seleccionar las métricas a utilizar en la evaluación marcando las deseadas.
7. Una vez configurada la evaluación, simplemente hay que pulsar el botón *Evaluar*. En pocos segundos, los resultados de la evaluación se mostrarán en la columna destinada a tal propósito, tal y como se aprecia en la figura 23. Al finalizar el proceso de evaluación, EvALL indicará mediante un mensaje *pop-up* el éxito o fracaso de las distintas evaluaciones, invitando al usuario a analizar el proceso en la consola PyEvALL. Como se puede ver en la imagen, EvALL 2.0 muestra los resultados en forma

EXIST 2023 Prueba 5

Evaluaciones ▾ Editar

EVALL FORM

Archivos de predicciones *

Examinar... No se han ... archivos.

Archivos de predicciones, puedes subir tantos como quieras.

Métricas

☐ Accuracy (Acc)

☐ Precision (P)

☐ Recall (R)

☐ FMeasure (F1)

☐ System Precision (SP)

☐ Cohen's Kappa (Kappa)

Without results

There are currently no results available. To evaluate one or more systems, upload the prediction files and select the metrics you want to evaluate with in the left side panel.

Figura 22: Interfaz del *dashboard* principal de EvALL 2.0.

de tabla, donde cada fila es un archivos de predicción y cada columna es una métrica seleccionada. El usuario puede ordenar los elemento de dicha tabla, tanto en orden ascendente como descendente, en base a cada una de las métricas analizadas.

ODESIA

Inicio Mis evaluaciones Repositorio FAQ Perfil Idioma

EVALL FORM

Archivos de predicciones *

Examinar... 6 archivos...cionados.

Archivos de predicciones, puedes subir tantos como quieras.

Métricas

☒ Accuracy (Acc)

☒ Precision (P)

☒ Recall (R)

☒ FMeasure (F1)

☐ System Precision (SP)

☐ Cohen's Kappa (Kappa)

☐ Information Contrast model (ICM)

☒ Information Contrast model Normalised (ICM Norm)

☐ Information Contrast Model Soft

RESULTS TABLE

	Output filename	Acc ↕	P ↕	R ↕	F1 ↕
▼	task1_Alex_P_UPB_1.json	0.80298838	0.69756780	0.80250969	0.74623266
▼	task1_CNLP-NITS-PP_1.json	0.70392916	0.61505664	0.70684118	0.65495687
▼	task1_IJIT SURAT_1.json	0.70282236	0.61264907	0.70449930	0.65360772
▼	task1_iimasGIL_NLP_2.json	0.82235750	0.71526705	0.82348173	0.76460000
▼	task1_InsightX_3.json	0.66685113	0.62722578	0.64320756	0.58394970
▼	task1_AIT_FHSTP_1.json	0.83010515	0.72150648	0.83062458	0.77166953

« 1/1 »

Gráficas

Metrics

Metrics

Metrics

Figura 23: Interfaz del *dashboard* principal EvALL 2.0 con resultados.

- Una vez la evaluación ha terminado y los resultados se han mostrado en la tabla, el usuario podrá acceder a la visualización de la consola PyEvALL para analizar el informe resumido de los éxitos y errores de la evaluación, tal y como se en la figura 24. Además, el usuario puede acceder al informe detallado que PyEvALL genera para cada archivo 25. Para ello, debe pulsar el botón *View* disponible en las opciones extra de cada fila de la tabla.
- Finalmente, el usuario podrá analizar los resultados de una forma visual mediante gráficas generadas automáticamente con los resultados obtenidos (ver figura 26). Así mismo, el usuario podrá realizar capturas de pantallas de las mismas para su posterior uso en artículos científicos, memorias técnicas, etc.

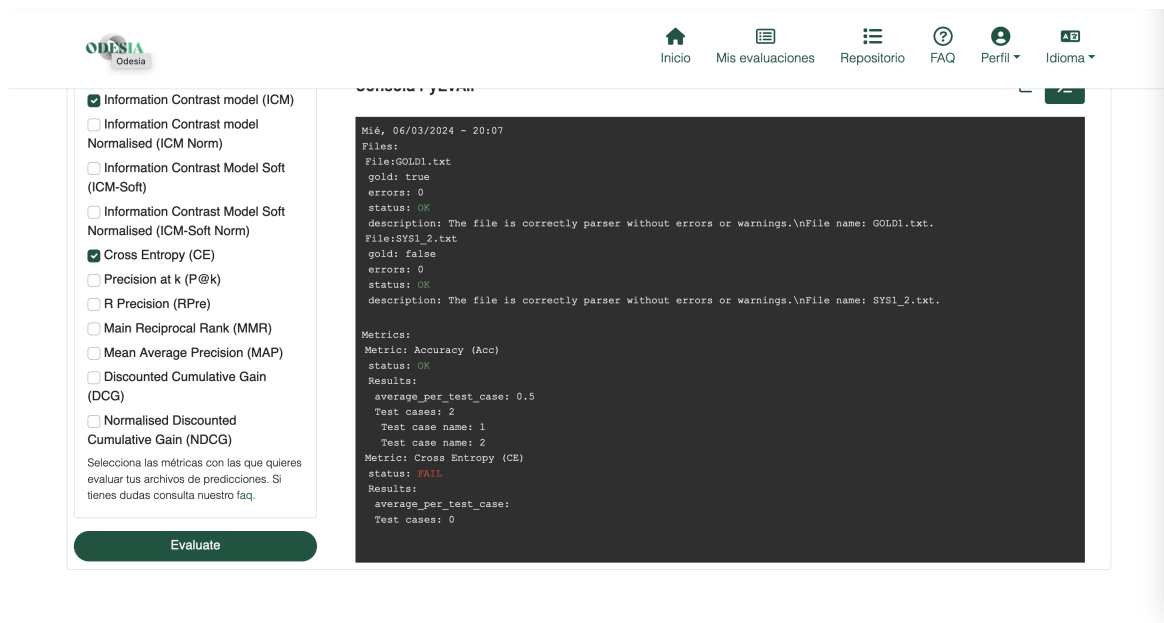


Figura 24: Interfaz del dashboard con la consola PyEvALL.

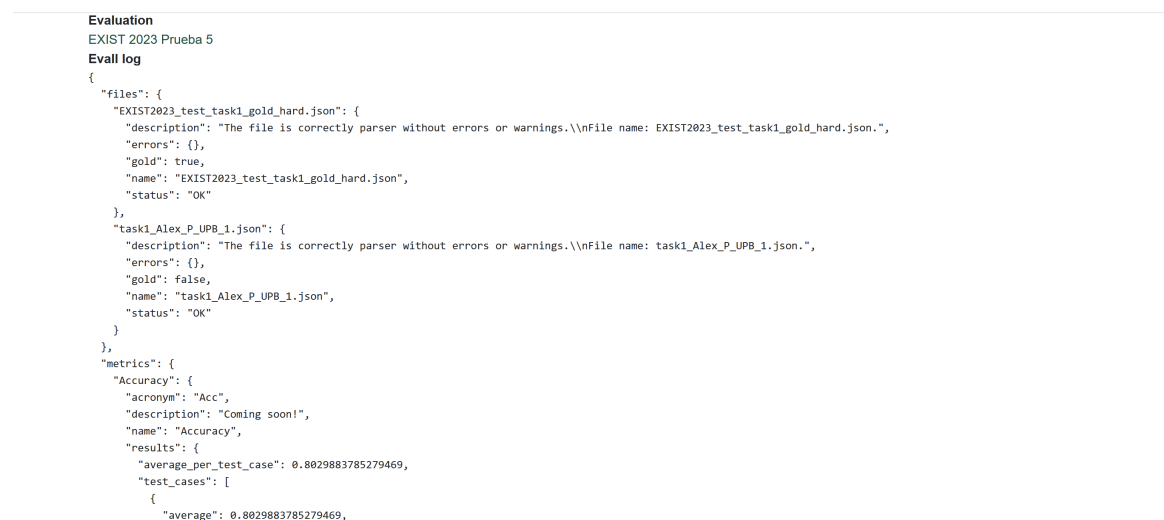


Figura 25: Interfaz del dashboard con el informe detallado de PyEvALL para un archivo de predicción.

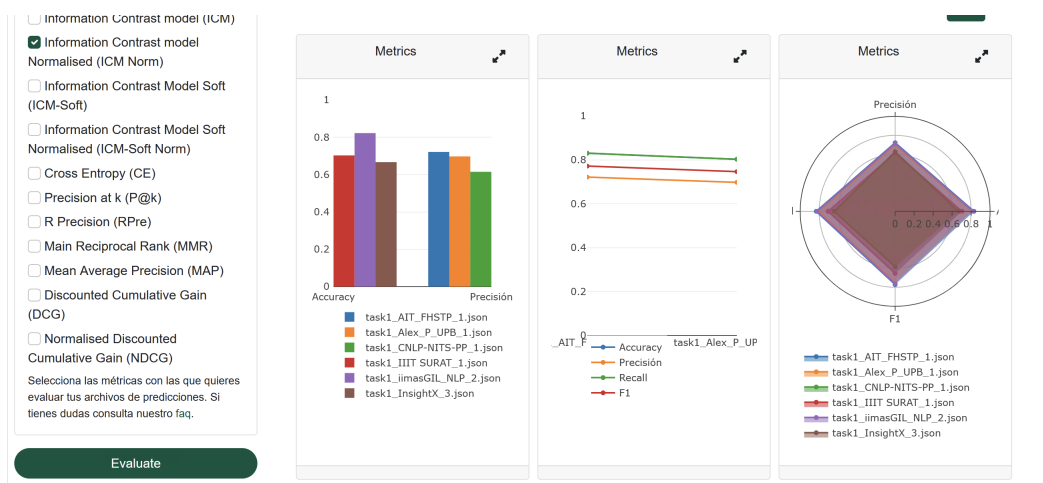


Figura 26: Interfaz del dashboard con gráficas.

8. Evaluación con *gold standard* desde repositorio

Esta sección describe el flujo de evaluación de EvALL 2.0, que proporciona soporte a investigadores o desarrolladores para la evaluación de sus sistemas de información usando un *gold standard* de los incluidos en el repositorio EvALL 2.0. En esta modalidad de evaluación, el usuario simplemente tiene que proporcionar los archivos con las predicciones de los modelos. A continuación, se detallan las funcionalidades del flujo, los casos de uso y su lógica, mostrando en cada paso las pantallas implementadas para ello. Dado que el desarrollo de este flujo es prácticamente similar al *Flujo de evaluación proporcionando el gold standard* (Sección 7), a lo largo de la descripción de este flujo se omitirán muchos contenidos repetidos.

8.1. Funcionalidad

Al igual que en el *Flujo de evaluación proporcionando el gold standard* (Sección 7), el objetivo de este flujo de evaluación es dotar a los usuarios de EvALL 2.0 de una herramienta que permita acceder a un conjunto amplio de contextos de evaluación y métricas. Sin embargo, esta evaluación se ha simplificado para que el usuario solo tenga que subir los archivos de predicciones. Esto tiene un doble beneficio: por un lado, el usuario no tiene que preocuparse de estar subiendo y formateando el *gold standard*, y por otro, favorece el avance del estado del arte sobre un entorno seguro y libre de posibles contaminaciones. Es decir, aquellas personas que dispongan de un *gold standard* y quieran compartirlo, pero no lo hagan para evitar contaminación de los actuales modelos generativos que constantemente *crawlean* la red, pueden subir el *gold standard* al repositorio EvALL 2.0, donde estará disponible para que los usuarios puedan evaluar sus modelos sin que nadie acceda al conjunto etiquetado de prueba. Por ello, en este flujo se propone una interfaz mediante la cual un usuario puede realizar una evaluación en cuatro simples pasos: (i) seleccionar el *gold standard* del repositorio; (ii) subir uno, o varios, archivos con predicciones; (iii) seleccionar las métricas deseadas; (iv) pulsar el botón *Evaluar*.

8.2. Casos de uso

Para la realización de este flujo, la plataforma solo contempla opciones para usuarios registrados.

8.2.1. Usuario - Registrado

El rol de usuario registrado tiene pleno acceso al flujo de evaluación proporcionando el *gold standard*, tal y como sucede en *Flujo de evaluación proporcionando el gold standard*. En realidad, la casuística de este flujo es exactamente igual que la del flujo de la sección 7, con la salvedad de que el usuario elige el *gold standard* del repositorio en lugar de proporcionarlo él mismo:

- **Visualizar los accesos al flujo de evaluación proporcionando un *gold standard*:** véase sección 7.
- **Acceder al dashboard de evaluación:** véase sección 7.
- **Realizar una evaluación:** véase sección 7.
- **Comparar resultados de varios modelos:** véase sección 7.
- **Eliminar un resultado de la tabla:** véase sección 7.
- **Visualizar el informe PyEvALL para ese archivo de predicción:** véase sección 7.
- **Visualizar la consola PyEvALL:** véase sección 7.

8.3. Lógica del flujo

La lógica de este flujo solo es accesible para usuarios registrados. A continuación, se describe la lógica del flujo paso a paso, asumiendo que el usuario ya está registrado, para realizar una evaluación con *gold standard* desde repositorio. La secuencia de pasos para este flujo es exactamente igual que la del flujo anterior (ver sección 7), con la salvedad de que en el formulario de creación de una evaluación el usuario debe elegir la opción *Evaluar utilizando un gold standard desde repositorio*.

1. En primer lugar el usuario registrado debe acceder a la interfaz de gestión de evaluaciones y pulsar el botón *Nueva Evaluación*, tal y como se explica en la sección 7.
2. Una vez en el formulario de creación de evaluaciones, el usuario debe seleccionar la opción *Evaluar utilizando un gold standard desde repositorio*, tal y como se observa en la figura 27.

Figura 27: Interfaz de EvALL 2.0 para crear una nueva evaluación desde repositorio.

3. Una vez seleccionada esta modalidad, el usuario debe buscar el *gold standard* contra el que se quiere evaluar. Para ello, puede comenzar a teclear y la aplicación irá sugiriendo posibles *gold standards* con coincidencias, tal y como se puede ver en la figura 28. En este caso concreto, el usuario ha comenzado a escribir el término *exi* y la aplicación le ha sugerido 4 gold standards de diferentes competiciones de la campaña de evaluación EXIST disponibles en el repositorio de EvALL 2.0.

Figura 28: Interfaz de EvALL 2.0 para crear una nueva evaluación desde repositorio: sugerencias EvALL 2.0.

4. El resto de pasos de este flujo son exactamente igual que los descritos en la sección 7, por lo que no se vuelven a incluir.

9. Librería de evaluación PyEvALL

PyEvALL es una herramienta de evaluación para sistemas de información que permite calcular un conjunto extenso de métricas que abarcan multitud de contextos, como clasificación o clustering. PyEvALL está implementada en Python y está basada en la teoría de la medida, recogiendo así las buenas prácticas en el área de evaluación de sistemas de información. El objetivo de PyEvALL es dotar a la comunidad científica, así como a otros actores en el área del desarrollo de sistemas de información, de una herramienta de evaluación de referencia que cubra multitud de contextos de evaluación y proporcione un amplio abanico de métricas.

Para conseguir este objetivo, se ha analizado el estado del arte en el área de evaluación de tareas de clasificación, clasificación ordinal, ranking, ranking con diversidad y clustering. Para cada uno de los contextos de evaluación, se han identificado las métricas del estado del arte, así como analizado sus debilidades y fortalezas de acuerdo a diversas propiedades formales, proponiendo en algunos casos nuevas métricas que abordan las limitaciones del estado del arte (Amigó et al., 2018; Amigó and Mizzaro, 2020; Amigo et al., 2020; Amigó et al., 2023; Amigo and Delgado, 2022). Este análisis, cuyo contenido y conclusiones escapan al ámbito de este informe, servirá como punto de partida para diseñar e implementar el conjunto de métricas seleccionadas del estado del arte, así como las nuevas métricas propuestas, y ponerlas a disposición de la comunidad científica.

Para que la comunidad pueda hacer un correcto uso de la librería, en todos sus diversos y posibles casos de uso, PyEvALL será liberada como código abierto una vez se disponga de una versión estable de la misma. De esta manera, se conseguirá, por un lado, una mayor difusión y acceso a la herramienta; y por otro, un mayor apoyo por parte de la comunidad para su desarrollo y detección de posibles errores.

9.1. Requisitos

Como ya se ha comentado, el objetivo fundamental de PyEvALL es convertirse en una herramienta de evaluación de referencia para la comunidad, para lo cual uno de sus principales requisitos es que sea de propósito general. Es decir, PyEvALL debe cubrir el **mayor número de contextos de evaluación y métricas posibles**. Para ello, su diseño debe aportar flexibilidad y modularidad, ya que la evaluación de sistemas de información es un área en constante cambio y actualización.

Además, debe tener una **curva de aprendizaje relativamente fácil** para poder llegar al máximo número de usuarios. En general, esto es una tarea complicada en este tipo de aplicaciones, pero dos de los puntos clave para conseguir una herramienta transparente para el usuario es el formato de entrada y la forma de ejecución. Respecto al primer punto, PyEvALL contará con un **formato unificado en json** a través del cual, y con los mismos atributos, se podrán ejecutar múltiples contextos y múltiples métricas, incluso métricas de diferentes contextos. Por otro lado, en contextos más complejos, como puede ser la clasificación multi-label con jerarquía y desacuerdo entre anotadores, el formato json aporta la flexibilidad y explicabilidad necesaria para abordar este tipo de contextos, o futuros, añadiendo los atributos necesarios. Además, PyEvALL dispondrá de distintos *wrappers* que permitan convertir formatos comúnmente utilizados por la comunidad científica, como TSV o *Trec_eval*, a formato json de una forma transparente para el usuario.

Respecto al segundo punto, PyEvALL estará disponible como un **paquete Python que podrá ser ejecutado de forma sencilla** mediante un método con tres parámetros obligatorios: ruta al archivos de predicciones, ruta al archivo de *gold standard* y lista con las métricas a ejecutar. Así mismo, al estar disponible como paquete Python, PyEvALL podrá ser utilizado desde entornos de programación de una forma sencilla y transparente para el desarrollador. Sin embargo, PyEvALL también debe ser capaz de dar soporte a usuarios más experimentados y con requisitos más exigentes. Para ello, la librería permite la evaluación de múltiples archivos de predicciones contra un *gold standard*, la obtención de un *dataframe* de resultados para su posterior análisis estadístico, o la modificación de múltiples parámetros, como pueden ser el formato de entrada, el formato de salida, los parámetros de las métricas utilizadas, entre otros, mediante el sistema de parametrización opcional de diccionarios de Python.

Otro de aspecto fundamental para su correcta implantación es la precisión, es decir, **dotar a la**

comunidad de una herramienta precisa y confiable. Para ello, PyEvALL está basado en la teoría de la medida, por lo que todas las métricas comprueban sus pre-condiciones, en caso de existir, antes de su ejecución, deteniendo la misma en caso de no cumplirse y explicando al usuario detalladamente los errores ocurridos. Además, la librería realiza un análisis exhaustivo de los datos de entrada, del *gold standard* y de las predicciones, para identificar posibles errores, parando la ejecución cuando sea necesario e informado al usuario. Por último, cada métrica incluida ha sido comparada contra otra implementación independiente de la misma bajo un conjunto extenso de casos de uso especialmente diseñados para cada contexto de evaluación.

Por último, PyEvALL debe proporcionar múltiples formatos de salida dependiendo de las necesidades del usuario, desde texto a tablas (dataframe o tsv). Para ello, se ha diseñado un formato único en json que podrá ser interpretado por diferentes sub-componentes y que conviertan dicha salida en informes con diferentes formatos.

9.1.1. Diseño y arquitectura

El desarrollo de la herramienta PyEvALL se ha realizado sobre el lenguaje de programación Python, utilizando el paradigma de programación orientado a objetos. A lo largo de las siguientes secciones, se mostrará la descripción a alto nivel de la arquitectura de PyEvALL, así como de sus diferentes componentes.

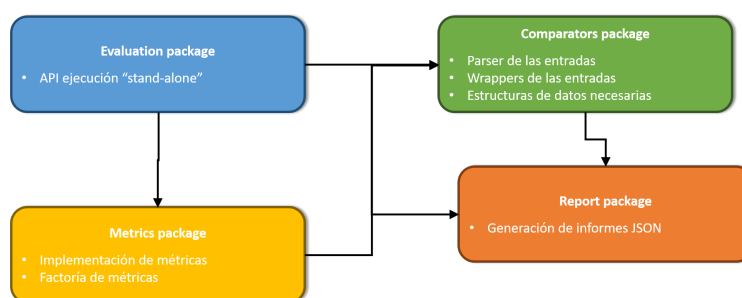


Figura 29: Arquitectura propuesta para el desarrollo de la librería PyEvALL.

Como se puede ver en la figura 29, la arquitectura está compuesta por cuatro paquetes principales:

- **Evaluation Package:** este paquete es el encargado de realizar las operaciones a alto nivel, ejecutando las distintas evaluaciones solicitadas, procesando los parámetros y convirtiendo los resultados al formato solicitado por el usuario. En realidad, este paquete hace las funciones de API a alto nivel de la librería. Mediante este paquete, como ya se ha indicado, se puede ejecutar la evaluación sencilla en la que se proporciona un archivo de predicciones, un archivo *gold standard* y una lista de métricas, obteniendo como resultado un informe en el formato deseado (json, json *embedded* con información textual detallada de cada error o *dataframe*). Además, PyEvALL permite otro método de evaluación en el que se proporciona una lista de predicciones, un *gold standard* y una lista de métricas, devolviendo un informe combinado que contiene todos los informes por pares.
- **Comparators Package:** este paquete realiza el análisis de los archivos de entrada, detectando posibles errores, tanto en las predicciones como en el *gold standard*. Así mismo, se propone la utilización de un único modelo de datos sobre el que trabajarán las distintas métricas con independencia de su formato, *PyEvALLComparator*, y que generarán en función de las demandas del usuario. Esta generalización de la estructura interna de datos de PyEvALL nos permitirá, a su vez, mejorar la eficiencia de la herramienta a la hora de realizar múltiples evaluaciones sobre diferentes métricas. Por ejemplo, la generación de la matriz de confusión para el cálculo de las métricas de clasificación se realizará una sola vez, y se usará en todas aquellas métricas la precisen. Este paquete, a su vez, incluye el módulo que analiza los archivos de entrada, identificando los errores y convirtiéndolos en comparadores de PyEvALL.

- **Metrics Package:** este paquete contiene la factoría de métricas, que permite crear los objetos métricas en tiempo de ejecución de acuerdo con la configuración de evaluación del usuario. Además, este modulo permite obtener una lista de todas las métricas disponibles en PyEvALL. Así mismo, este paquete incluye el modulo *PyEvALLMetric*, donde se incluye la implementación de todas las métricas, independientemente de su contexto de evaluación, gracias a su diseño genérico.
- **Report Package:** este paquete se encarga de generar el informe PyEvALL interno, en formato json, resultante de analizar los formatos de entrada, comprobar las pre-condiciones de las métricas y ejecutar todas aquellas que no han generado errores. Este informe interno de PyEvALL es la entrada para el resto de informes como el informe *PyEvALLEmbeddedReport*, que incluye las descripciones en texto plano de los distintos errores y *warnings* que el sistema va encontrándose, o el informe *PyEvALLDataframeReport*, que genera un *dataframe* donde cada fila es un archivo de predicciones y cada columna una métrica.

Cada componente está, a su vez, formado por diferentes clases que forman una estructura jerarquizada y dotan a cada paquete de la funcionalidad requerida para alcanzar los objetivos y funcionalidades descritos. Como se puede apreciar en el diseño de arquitectura propuesto, se trata de un diseño sencillo, pero que haciendo uso de la herencia y abstracción proporcionadas por el paradigma de orientación a objetos, posibilita futuras actualizaciones de una forma sencilla y modular.

9.2. Funcionalidad de la librería

Como ya se ha descrito, la funcionalidad principal de PyEvALL es permitir la evaluación del par [archivo de predicciones, archivo de *gold standard*] sobre un conjunto de métricas, independientemente del contexto de evaluación. En particular, el contexto de evaluación es transparente para el usuario y viene dado por dos factores: (i) **el formato de los archivos de entrada**, que es un aspecto clave ya que, por ejemplo, para evaluar ranking es necesario un valor numérico mientras que para evaluar en un contexto multi-label de clasificación se necesita un vector de clases; (ii) **las métricas de evaluación**, que son válidas en un conjunto limitado de contextos: F1, por ejemplo, es válida para clasificación mono-label y multi-label pero no para el paradigma LeWiDi (Learning with Disagreement). Con estos dos elementos, PyEvALL es capaz de determinar automáticamente si la evaluación puede realizarse de forma satisfactoria para cada métrica y cada *test case* del archivo de predicciones, informando al usuario del error en caso negativo, o mostrando los resultados en caso afirmativo.

Actualmente, PyEvALL dispone de métricas implementadas para contemplar los siguientes contextos de evaluación:

- **Clasificación mono-label sin jerarquía:** contexto de evaluación en el que a cada instancia se le asigna una clase objetivo, y solo una. Además, las clases no tienen orden o jerarquía entre ellas y todas tienen la misma relevancia.
 - **Accuracy:** se define simplemente como la proporción de respuestas correctas.
 - **System Precision:** se define como la proporción de respuestas correctas con respecto a los elementos que aparecen en la salida del sistema.
 - **Kappa:** el estadístico Kappa de Cohen, una versión normalizada de la accuracy que utiliza la puntuación de accuracy hipotética de un sistema no informativo que asigna etiquetas aleatorias a los elementos, pero manteniendo la misma distribución de etiquetas en la salida del sistema evaluado.
 - **Precision:** esta métrica estima la probabilidad de encontrar elementos bien clasificados en la clase de salida del sistema.
 - **Recall:** esta métrica estima la probabilidad de clasificar correctamente los elementos en el *gold standard*.
 - **FMeasure:** es una media armónica ponderada de *precision* y *recall*. La propiedad más importante de esta función es que penaliza las puntuaciones bajas tanto en Precision como en Recall (más duramente que la media aritmética).

- **Information Contrast Model:** es una función de similitud que generaliza la información puntual mutua (PMI) y puede utilizarse para evaluar los resultados del sistema en problemas de clasificación calculando su similitud con las categorías del *gold standard*.
- **Information Contrast Model Normalizado:** la versión normalizada de ICM con el valor máximo y mínimo del *gold standard*.
- **Clasificación multi-label sin jerarquía:** contexto de evaluación en el que a cada instancia se le asigna una clase, o varias, de entre un conjunto de clases objetivo. Además, las clases no tienen orden o jerarquía entre ellas y todas tienen la misma relevancia.
 - **Precision:** esta métrica estima la probabilidad de encontrar elementos bien clasificados en la clase de salida del sistema, pero en este caso sobre salidas con múltiples opciones.
 - **Recall:** esta métrica estima la probabilidad de clasificar correctamente los elementos en el *gold standard*, pero en este caso sobre salidas con múltiples opciones.
 - **FMeasure:** es una media armónica ponderada de *precision* y *recall*. La propiedad más importante de esta función es que penaliza las puntuaciones bajas tanto en *precision* como en *recall* (más duramente que la media aritmética). De nuevo, en este caso sobre salidas con múltiples opciones.
- **Clasificación mono-label jerárquica:** contexto de evaluación en el que a cada instancia se le asigna una clase objetivo, y solo una. Además, las clases tienen una relación jerárquica, de forma que los errores entre clases del mismo nivel jerárquico representan un fallo menor que los errores entre clases de niveles jerárquicos diferentes.
 - **Information Contrast Model:** es una función de similitud que generaliza la información puntual mutua (PMI) y puede utilizarse para evaluar los resultados del sistema en problemas de clasificación calculando su similitud con las categorías del *gold standard*.
 - **Information Contrast Model Normalizado:** la versión normalizada de ICM con el valor máximo y mínimo del *gold standard*.
- **Clasificación multi-label jerárquica:** contexto de evaluación en el que a cada instancia se le asigna una clase, o varias, de entre un conjunto de clases objetivo. Además, las clases tienen una relación jerárquica, de forma que los errores entre clases del mismo nivel jerárquico representan un fallo menor que los errores entre clases de niveles jerárquicos diferentes.
 - **Information Contrast Model:** es una función de similitud que generaliza la información mutua puntual (PMI) y puede utilizarse para evaluar los resultados del sistema en problemas de clasificación calculando su similitud con las categorías del *gold standard*.
 - **Information Contrast Model Normalizado:** la versión normalizada de ICM con el valor máximo y mínimo del *gold standard*.
- **LeWiDi - clasificación mono-label:** contexto de evaluación en el que cada instancia tiene una distribución de probabilidades para todas las clases posibles y donde cada etiqueta solo ha podido ser asignada una vez por cada anotador, por lo que la suma de las probabilidades de todas las etiquetas tiene que ser 1. Además, las clases tienen una relación jerárquica, de forma que los errores entre clases del mismo nivel jerárquico representan un fallo menor que los errores entre clases de niveles jerárquicos diferentes. Las métricas incluidas actualmente en este contexto son:
 - **Information Contrast Model Soft:** es una función de similitud que generaliza la información puntual mutua (PMI), adaptada para trabajar con distribuciones de probabilidades, y que puede utilizarse para evaluar los resultados del sistema en problemas de clasificación calculando su similitud con las categorías del *gold standard*.
 - **Information Contrast Model Soft Normalizado:** la versión normalizada de ICM Soft con el valor máximo y mínimo del *gold standard*.

- **Cross Entropy:** es una función de pérdida que puede utilizarse para cuantificar la diferencia entre dos distribuciones de probabilidad.
- **LeWiDi - clasificación multi-label jerárquica:** contexto de evaluación en el que cada instancia tiene una distribución de probabilidades para todas las clases posibles y para cada instancia se asigna una clase, o varias, de entre un conjunto de clases objetivo. Además, las clases tienen una relación jerárquica, de forma que los errores entre clases del mismo nivel jerárquico representan un fallo menor que los errores entre clases de niveles jerárquicos diferentes.
 - **Information Contrast Model Soft:** es una función de similitud que generaliza la información puntual mutua (PMI), adaptada para trabajar con distribuciones de probabilidades, y que puede utilizarse para evaluar los resultados del sistema en problemas de clasificación calculando su similitud con las categorías del *gold standard*.
 - **Information Contrast Model Soft Normalizado:** la versión normalizada de ICM Soft con el valor máximo y mínimo del *gold standard*.
- **Ranking:** en el contexto de evaluación de ranking, las métricas tienen como objetivo cuantificar en qué medida un ranking producido por los sistemas es compatible con los valores de relevancia asignados en el *gold standard*. En la mayoría de los casos, las tareas de ranking se evalúan en colecciones que contienen una gran cantidad de documentos. Para fines de evaluación, EvALL considera que cada documento no recuperado por el sistema se asume como irrelevante.
 - **Precision at K:** supone que el usuario explora sólo las k primeras posiciones de la clasificación.
 - **R Precision:** la métrica mide la precisión de la clasificación de salida del sistema en el momento en que se alcanza un determinado recall. Supone que el usuario no detiene la búsqueda hasta encontrar una determinada proporción de elementos relevantes con respecto a los elementos relevantes anotados en el *gold standard*.
 - **Main Reciprocal Rank:** MMR sólo tiene en cuenta la posición del primer documento relevante en la clasificación. Se recomienda cuando el usuario sólo está interesado en encontrar un elemento relevante, y podemos suponer que cualquier elemento anotado como relevante en el *gold standard* satisfará las necesidades del usuario.
 - **Mean Average Precision:** MAP ha sido una de las métricas más populares en tareas de ranking. Es adecuada cuando el usuario mantiene la exploración hasta encontrar una cantidad representativa de documentos relevantes.
 - **Discounted Cumulative Gain:** esta métrica es adecuada cuando el usuario explora niveles profundos de la clasificación de resultados del sistema, y los nuevos documentos relevantes aumentan su utilidad de forma aditiva, cercana a una función lineal.
 - **Normalized Discounted Cumulative Gain:** dado que los resultados según DCG pueden variar en tamaño entre diferentes consultas o sistemas, la versión normalizada de DCG utiliza una puntuación DCG ideal (IDCG) para comparar los resultados.

La implementación de cada métrica se ha realizado siguiendo la base teórica de la teoría de la medida, identificando, cuando procede, las pre-condiciones de la misma y evaluando su veracidad o no antes de ejecutarse. Cada métrica ha sido comprobada contra una implementación independiente con un conjunto de casos de prueba desarrollados para tal propósito. Por último, PyEvALL permite realizar evaluaciones con métricas de diferentes contextos de evaluación, pudiendo, por ejemplo, realizar una evaluación que incluya la métrica *Accuracy*, utilizada en el contexto de clasificación mono-label sin jerarquía, a la vez que ICM con jerarquía, utilizada en un contexto de clasificación mono-label con jerarquía. PyEvALL permite también evaluaciones personalizadas mediante la parametrización de aquellas métricas que lo precisan de acuerdo a las necesidades del usuario. Para ello, los parámetros específicos deben ser incluidos como parámetros en la configuración de la evaluación. En caso de no encontrarse esos parámetros en la configuración, la librería utilizará los parámetros por defecto.

Una vez realizada la evaluación de cada métrica seleccionada, PyEvALL produce un informe de evaluación en formato json. En este informe también se recogen los posibles errores o alertas detectados en el análisis de los archivos de entrada o el análisis de las pre-condiciones, informando de cada uno al usuario. En concreto, y respecto al análisis de los archivos de entradas y sus formatos, se comprueba si la ruta del archivo existe, si el archivo está vacío, si es un formato json correcto, si hay identificadores de instancias duplicadas, si faltan valores en los atributos, o si el número de atributos es incorrecto. Nótese que estos errores detectados producen alertas que informan al usuario si son encontrados en el archivo de predicciones, permitiendo continuar con la evaluación si fuera posible. Por el contrario, estos errores producen la detención de la evaluación impidiendo la ejecución del cálculo de la métrica, e informando al usuario, si son detectados en el *gold standard*. Por último, el usuario puede elegir obtener el informe final en formato json por defecto, en un formato json enriquecido con descripciones textuales de los errores detectados o en un informe con formato *dataframe* con el que poder procesar y realizar nuevos experimentos o estadísticas.

9.3. Formatos de entrada y salida

Como ya se ha expuesto, el formato principal de entrada y salida de PyEvALL es el formato json. A continuación, se detalla el formato PyEvALL, describiendo sus características y atributos. Así mismo, se presenta el *wrapper* disponible en la librería para formatos TSV y CSV.

9.3.1. Formato de entrada PyEvALL

El formato de entrada para PyEvALL es único para todos los contextos de evaluación, y está encapsulado en un formato json. La elección de este formato se ha basado principalmente en la gran versatilidad que proporciona, así como en la facilidad para controlar posibles errores. Gracias a esto, se han podido definir los formatos que permiten trabajar con todos los contextos de evaluación definidos en la sección anterior 9.2. Nótese que lo normal en este tipo de herramientas es que, en cada contexto, se suela tener un formato diferente de acuerdo a las necesidades del mismo. En nuestro caso, con el mismo formato se pueden evaluar diferentes contextos, aunque contextos más complejos requieren de alguna variación del formato básico, como se verá a continuación.

Además, el formato json permite, a su vez, la declaración de un *schema*, como en otros formatos estructurados como XML, mediante el cual se pueden validar fácilmente los datos de entrada. Por este motivo, se ha creado un *schema* para el formato de entrada de PyEvALL donde se definen los atributos necesarios, así como sus tipos y sus características, y que es el siguiente (ver figura 30):

```
FORMAT_JSON_SCHEMA= {
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "test_case": {"type": "string"},
      "id": {"type": "string"},
      "value": {
        "anyOf": [
          {"type": "string"},
          {"type": "array", "items": {"type": "string"}},
          {"type": "integer"},
          {
            "type": "object",
            "patternProperties": {
              "^[a-zA-Z0-9_]+$": {"type": "number"},
            }
          }
        ],
      },
    },
  },
  "required": ["test_case", "id", "value"],
  "additionalProperties": false
}
```

Figura 30: Formato json de la librería PyEvALL.

Como se puede ver en la imagen 30, el *schema* del formato json de PyEvALL es un vector de objetos json, cada uno representando una instancia de evaluación, donde cada objeto está compuesto por tres elementos:

- **test_case:** es el identificador unívoco del conjunto de datos. En clasificación, este valor suele ser único para todo el dataset, aunque podría usarse para realizar experimentos sobre diferentes versiones

de datasets y calcular relevancias estadísticas (por ejemplo, EXIST 2021, 2022 y 2023). En ranking, este valor suele ser representativo de las consultas o búsquedas realizadas para cada termino en los datasets.

- **id:** este atributo representa el valor unívoco de cada *item*, y admite tanto formato string o int, teniendo que ser el mismo para el archivo de predicciones y de *gold standard*. Este valor debe ser único para cada elemento y no puede estar repetido, salvo en ciertas excepciones. La presencia de elementos repetidos en los distintos archivos produce diferentes efectos. Un elemento repetido en el *gold standard* provoca la interrupción de la evaluación, mientras que un error de ids duplicados en las predicciones provoca un *warning*, indicando que los elementos duplicados identificados después del primero serán ignorado.
- **value:** este campo representa el valor asignado a cada *item* y cuyo tipo variará dependiendo del contexto de evaluación aplicado. Por ejemplo, para clasificación mono-label, el elemento estará compuesto por un *string*, mientras que para clasificación multi-label, el elemento estará compuesto por un vector de *strings*.

Tal y como indica en el *schema*, los tres atributos de cada objeto json son necesarios para el correcto funcionamiento de PyEvALL, por lo que su ausencia provocará un error.

9.3.1.1. Formato para clasificación mono-label

Este formato es el formato típico de las tareas de clasificación mono-label, donde cada *item* tiene una única clase asociada. En este formato, la etiqueta será cualquier cadena de caracteres que representa una clase objetivo. Un ejemplo correcto de este contexto de evaluación en formato json PyEvALL sería el mostrado en la figura 31.

```
[
  {
    "test_case": "EXIST2023",
    "id": "I1",
    "value": "A"
  },
  {
    "test_case": "EXIST2023",
    "id": "I2",
    "value": "B"
  },
  {
    "test_case": "EXIST2023",
    "id": "I3",
    "value": "C"
  }
]
```

Figura 31: Formato json para clasificación mono-label de la librería PyEvALL.

En el ejemplo mostrado, se puede ver que el vector está formado por tres elementos pertenecientes a un mismo *test_case*, *EXIST2023*, con tres identificadores diferentes (“I1”, “I2” e “I3”), y tres clases objetivo diferentes (“A”, “B” y “C”).

9.3.1.2. Formato para clasificación multi-label

El formato de clasificación multi-label es aquel en el que cada *item* puede estar clasificado con una o varias clases objetivo. Por este motivo, el formato en PyEvALL de este contexto está compuesto de los mismos elementos que en el caso anterior, con la diferencia de que ahora el atributo “value” es un vector de elementos. Estos elementos, a su vez, deben ser cadenas de caracteres que representan una clase objetivo. Un ejemplo correcto de este contexto de evaluación en formato json PyEvALL sería el mostrado en la figura 32.

Como se puede ver en el ejemplo, el archivo está formado por un vector de objetos json con tres elementos con los mismos campos que en el caso anterior, pero con la diferencia, en este caso, de que el atributo “value” esta compuesto por un vector con las clases objetivos de cada *item*.

9.3.1.3. Formato para LeWiDi

El formato de clasificación con desacuerdos permite asignar a cada elemento del dataset una distribución

```
[
  {
    "test_case": "EXIST2023",
    "id": "I1",
    "value": ["A"]
  },
  {
    "test_case": "EXIST2023",
    "id": "I2",
    "value": ["A", "B", "C", "D"]
  },
  {
    "test_case": "EXIST2023",
    "id": "I3",
    "value": ["C", "E"]
  }
]
```

Figura 32: Formato json para clasificación multi-label de la librería PyEvALL.

de probabilidades asociada a cada clase. Es decir, en lugar de seleccionar una única clase objetivo absoluta para cada *item*, la distribución de etiquetas por anotador es asignada a cada elemento. Para ello, como se puede ver en el siguiente ejemplo, el formato utilizado por PyEvALL es el mismo que en los casos anterior, con la salvedad de que el atributo “value” es representado con un diccionario Python donde cada elemento representa una clase objetivo y su valor representa la probabilidad asociada. Nótese que, en el caso de clasificación con desacuerdo mono-label, la suma para cada elemento debe ser 1, mientras que para clasificación multi-label no es necesario. Un ejemplo correcto de este contexto de evaluación en formato json PyEvALL sería el mostrado en la figura 33.

```
[
  {
    "test_case": "1",
    "id": "I1",
    "value": {
      "B": 0.6,
      "C": 0.4
    }
  },
  {
    "test_case": "1",
    "id": "I2",
    "value": {
      "B": 0.5,
      "C": 0.5
    }
  },
  {
    "test_case": "1",
    "id": "I3",
    "value": {
      "B": 0.9,
      "C": 0.1
    }
  }
]
```

Figura 33: Formato json para contexto de evaluación LeWiDi de la librería PyEvALL.

9.3.1.4. Formato Ranking

En el formato del contexto de evaluación ranking, cada *item* tiene asignado un valor indicando la posición de ordenación en el ranking, en el caso de las predicciones, y el valor de relevancia, en el caso del *gold standard*. Como se puede observar en el siguiente ejemplo, el formato es exactamente el mismo que en los casos anteriores, pero con la diferencia de que ahora el valor del atributo “value” está formado por números que representan, en cada caso, la ordenación o relevancia del *item*. Un ejemplo correcto de este contexto de evaluación en formato json PyEvALL sería el mostrado en la figura 34.

```
[
  {
    "test_case": "GOLD1",
    "id": "A",
    "value": 1
  },
  {
    "test_case": "GOLD1",
    "id": "B",
    "value": 2
  }
]
```

Figura 34: Formato json para contexto de evaluación LeWiDi de la librería PyEvALL.

Como se puede ver en la imagen, en este ejemplo se muestran las predicciones de un sistema de ranking que ha indicado que el *item* con identificador “A” tiene asignada la posición 1 del ranking, mientras que el *item* con el identificador “B” tiene asignada la posición 2 del ranking.

9.3.2. Formato de entrada para la jerarquía

Dado que la librería de PyEvALL permite evaluar contextos de evaluación con jerarquía, se hace preciso un formato para la misma, así como una parametrización opcional en el sistema. Para ello, se ha incluido el parámetro opcional *hierarchy*, que permite asociar una estructura de tipo diccionario de Python, tal y como se puede ver en la figura 35, donde las hojas son representadas por listas de elementos.

```
{
  "YES":
  [
    "IDEOLOGICAL-INEQUALITY",
    "STEREOTYPING-DOMINANCE",
    "OBJECTIFICATION",
    "SEXUAL-VIOLENCE",
    "MISOGYNY-NON-SEXUAL-VIOLENCE"
  ],
  "NO":
  []
}
```

Figura 35: Formato jerarquía de la librería PyEvALL.

Esta jerarquía puede proporcionarse directamente como *string* o como ruta a un archivo que contenga dicha jerarquía.

9.3.3. Wrapper de entrada para TSV

PyEvALL incluye un *wrapper* para uno de los formatos más utilizados en investigación, como es el formato tsv (del inglés *tab-separated values*), que se compone de tres columnas con cabeceras, tal y como se muestra en la figura 36, y donde cada columna representa cada uno de los atributos del formato json. El formato tsv es un formato abierto sin unos estándares fijos que permitan la creación de *schemas*, por lo que es un lenguaje más propenso a errores. Actualmente, en la versión disponible de PyEvALL, se han capturado muchos de los posibles errores, pero deben realizarse más pruebas para asegurar que todos los errores han sido capturados correctamente. Es importante resaltar que el formato tsv de PyEvALL precisa de las cabeceras. En caso de no existir éstas, producirá un *warning* y continuará con el procesamiento.

test_case	id	value
1	1	"TRUE"
1	2	"TRUE"
1	3	"TRUE"

Figura 36: Formato tsv de la librería PyEvALL.

De igual modo, y tal como se puede ver en la imagen 37, PyEvALL también dispone de un *wrapper* para el formato csv (del inglés *comma-separated-values*).

9.3.4. Formato del informe de evaluación json

Los informes de evaluación de PyEvALL son generados en formato json mediante la concatenación de diferentes diccionarios Python con información generada durante el proceso de evaluación. Esta composición de diferentes diccionarios genera un json genérico utilizado por PyEvALL para todo su proceso interno de evaluación. Como se ha expuesto anteriormente (ver sección 9.2), la evaluación en PyEvALL está diseñada sobre el par [*archivo de predicciones*, *archivo de gold standard*], y de igual manera, los informes se centran en este par.

```
testcase,id,value
5,1,A
5,2,A
5,3,A
5,4,B
5,5,B
5,6,B
5,7,D
5,8,B
6,7,C
6,4,A
6,5,D
```

Figura 37: Formato csv de la librería PyEvALL.

Un fragmento de un informe PyEvALL se puede ver en la imagen 38. El informe consta de dos elementos principales: las métricas, *metrics*, y los archivos, *files*.

En el elemento de métricas, se encuentran los atributos principales de la métrica, y que pueden ser útiles para generar informes de otro tipo, como puede ser el nombre o el acrónimo, así como los resultados de la misma. Además, este elemento incluye los posibles errores en el análisis de las pre-condiciones de cada métrica, si los hubiera. Por ejemplo, tal y como se ve en la figura 38, la métrica *precision* no ha cumplido con su pre-condición de formato de entrada para el par de archivos proporcionados, por lo que no se ha ejecutado y se informa de ello al usuario. Por otro lado, el elemento de archivos recoge los posibles errores detectados en los archivos analizados al realizar la evaluación, es decir, el archivo de predicciones y el archivo de *gold standard*. Cada elemento incluye, a su vez, una descripción de cada error o análisis, permitiendo la generación de informes más explicativos y completos. En concreto, para poder obtener las explicaciones textuales de los errores y el análisis del proceso de evaluación con explicaciones embebidas se usa el parámetro *reportembedded*.

Como se puede observar, el formato propuesto es sencillo y fácilmente extensible a nuevos atributos o información necesaria. Así mismo, el formato propuesto puede ser interpretado por diversos analizadores, proporcionando al usuario informes enriquecidos y adaptados a sus necesidades, como por ejemplo, el informe con explicaciones embebidas.

9.3.4.1. Formato de informe de evaluación con explicaciones embebidas

El informe con explicaciones embebidas de PyEvALL es un informe json con informaciones textuales embebidas que describen los errores y procesos de análisis realizados durante la evaluación. En él se detallan de forma clara y precisa los errores detectados en los archivos de entrada. Es un claro ejemplo de las grandes posibilidades que ofrece el informe PyEvALL en json, mediante cual se pueden hacer analizadores que extraigan la información y generar nuevos informes, ya sea en json como en otros formatos como tsv o latex.

Un ejemplo del formato de informe con explicaciones embebidas se puede ver en la figura ???. Tal y como se aprecia en la imagen, este informe es prácticamente igual al anterior con la salvedad de que el campo *description* incluye las descripciones. Volviendo al ejemplo anterior, la métrica *precision* genera un error dado que el formato de entrada no es apropiado para este contexto de evaluación, tal y como se explica en el mensaje. Así mismo, se indica que los archivos han sido procesados adecuadamente.

9.3.4.2. Formato de informe de evaluación de DataFrame

Por último, y orientado al análisis de varias métricas sobre varios archivos de predicciones, PyEvALL incluye el informe DataFrame. Este informe, como su propio nombre indica, esta formado por un *dataframe* de la librería Pandas de Python. El informe puede ser obtenido mediante código, para su posterior análisis, o impreso en formato tabla, para una mejor visualización (ver figura 40). En este caso, el informe mostrado en la imagen es un informe generado mediante el método de PyEvALL que permite realizar una evaluación dada una lista de archivos de predicciones sobre un conjunto de métricas. Como se puede ver en la imagen, este tipo de informes suelen ser de gran utilidad para un análisis en profundidad

```
{
  "metrics": {
    "Precision": {
      "name": "Precision",
      "acronym": "Pr",
      "description": "Use parameter: report=\\"embedded\\"!",
      "status": "FAIL",
      "results": {
        "test_cases": [],
        "average_per_test_case": null
      },
      "preconditions": {
        "METRIC_PRECONDITION_NOT_VALID_FORMAT_FOR_CONTEXT_EVALUATION": {
          "name": "METRIC_PRECONDITION_NOT_VALID_FORMAT_FOR_CONTEXT_EVALUATION",
          "description": "Use parameter: report=\\"embedded\\"!",
          "status": "FAIL",
          "test_cases": ["1"]
        }
      }
    },
    "ICMSoft": {
      "name": "Information Contrast Model Soft",
      "acronym": "ICM-Soft",
      "description": "Use parameter: report=\\"embedded\\"!",
      "status": "OK",
      "results": {
        "test_cases": [{
          "name": "1",
          "average": -4.555258212153537
        }],
        "average_per_test_case": -4.555258212153537
      }
    }
  },
  "files": {
    "SYS2_SOFT_P1.json": {
      "name": "SYS2_SOFT_P1.json",
      "status": "OK",
      "gold": false,
      "description": "Use parameter: report=\\"embedded\\"!",
      "errors": {}
    },
    "GOLD_SOFT_P1.json": {
      "name": "GOLD_SOFT_P1.json",
      "status": "OK",
      "gold": true,
      "description": "Use parameter: report=\\"embedded\\"!",
      "errors": {}
    }
  }
}
```

Figura 38: Formato del informe de evaluación json de la librería PyEvALL.

de diferentes modelos sobre diferentes métricas, algo muy frecuente en la comunidad científica.

Cabe destacar que, aunque es posible realizar este informe solo sobre el par [*archivo de predicciones*, *archivo de gold standard*], su principal utilidad es la comparación entre diferentes archivos de predicciones.

10. Trabajo Futuro

Durante este año, se ha realizado un gran avance en el desarrollo de la herramienta EvALL 2.0, disponiéndose actualmente de un prototipo fiable y en disposición de ser utilizado. No obstante, se han identificado nuevas funcionalidades necesarias de cara a conseguir el impacto deseado, tanto en la comunidad científica como empresarial, y que hagan de EvALL 2.0 el referente en el área que aspira a ser. Para ello, durante el siguiente año se pretende incorporar las siguientes funcionalidades:

- **Desarrollo del flujo para organización de campañas de evaluación:** mediante este flujo, los organizadores de campañas de evaluación dispondrán de una aplicación fiable y gratuita que les ayudaría en el proceso de evaluación de los resultados de los participantes.
- **Mejora de la interfaz de *dashboard* con nuevas funcionalidades:** aunque la interfaz propuesta es amigable y fácil de usar, se plantean muchas posibilidades, como puede ser una mejor visualización del informe detallado y de la consola de PyEvALL, la inclusión de más gráficas adaptadas a los contextos de evaluación, la mejora del panel de selección de métricas que incluya la parametrización de las mismas, etc.
- **Flujo para publicar un resultado:** mediante este flujo, EvALL 2.0 se convertirá en un referente del estado del arte, permitiendo a todos los usuarios publicar sus resultados en el repositorio EvALL 2.0.
- **Flujo para subir un *gold standard*:** mediante este flujo, el crecimiento de EvALL 2.0 estaría garantizado a lo largo de tiempo por parte de la comunidad.
- **Desarrollo e inclusión de nuevas métricas y contextos de evaluación:** durante el tercer año del proyecto ODESIA, se pretende aumentar el número de métricas en los contextos de evaluación ya existentes, así como añadir nuevos contextos.

11. Conclusiones

En este informe se ha presentado la versión 2 del portal web de evaluación EvALL 2.0 para sistemas de información. El portal ha sido desarrollado por el Grupo de Investigación en Procesamiento de Lenguaje Natural y Recuperación de Información de la UNED, en el marco del proyecto del Espacio de Observación de Inteligencia Artificial en Español, en concreto Ámbito 1 Estado del arte comparado, Actividad 1.3 - Aplicación web EvALL 2.0.

Durante este segundo año de trabajo, se ha realizado un trabajo importantísimo en el desarrollo de toda aplicación software de cierto calado, como es el diseño y desarrollo de un primer prototipo amigable y usable. El prototipo desarrollado facilita enormemente a la comunidad científica e industria uno de los procesos más importantes para el avance de la investigación como es la evaluación. Además, EvALL 2.0 lo hace de una forma transparente y didáctica para el usuario. En concreto, durante este año se ha rediseñado la apariencia de toda la aplicación y se ha mejorado diseño y desarrollo del dashboard, el centro neurálgico de EvALL 2.0. Este desarrollo incluye nuevas formas de visualizar y analizar los resultados, tanto textual (mediante los informes de PyEvALL), como visual (mediante las distintas gráficas generadas automáticamente en base a los resultados obtenidos). Además, se han incluido los flujos de gestión de evaluaciones, evaluación contra repositorio, desarrollo del repositorio EvALL 2.0 y la visualización del mismo. Respecto a la librería de evaluación PyEvALL, se han incluido tres nuevos contextos de evaluación y se han implementado diez nuevas métricas asociadas a dichos contextos y a otros ya incluidos. Por último, se ha mejorado el tratamiento de errores en los formatos de predicciones y *gold standard*, así como el análisis de las pre-condiciones de las métricas que lo precisan.

Por último, y aunque el avance en este año ha sido muy relevante, se han señalado los objetivos que guiarán el trabajo del año siguiente del proyecto, y que pretenden dotar a EvALL 2.0 de unas funcionalidades únicas.

12. Agradecimientos

Este trabajo ha sido financiado por la Unión Europea - NextGenerationEU a través del “Plan de Recuperación, Transformación y Resiliencia”, por el Ministerio de Asuntos Económicos y Transformación Digital y por la UNED. Sin embargo, los puntos de vista y las opiniones expresadas son únicamente los del autor o autores y no reflejan necesariamente los de la Unión Europea o la Comisión Europea. Ni la Unión Europea ni la Comisión Europea pueden ser consideradas responsables de los mismos.

Bibliografía

- Enrique Amigo and Agustín Delgado. 2022. [Evaluating extreme hierarchical multi-label classification](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5809–5819, Dublin, Ireland. Association for Computational Linguistics.
- Enrique Amigó, Julio Gonzalo, and Stefano Mizzaro. 2023. [What is my problem? identifying formal tasks and metrics in data mining on the basis of measurement theory](#). *IEEE Trans. Knowl. Data Eng.*, 35(2):2147–2157.
- Enrique Amigo, Julio Gonzalo, Stefano Mizzaro, and Jorge Carrillo-de Albornoz. 2020. [An effectiveness metric for ordinal classification: Formal properties and experimental results](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3938–3949, Online. Association for Computational Linguistics.
- Enrique Amigó and Stefano Mizzaro. 2020. [On the nature of information access evaluation metrics: a unifying framework](#). *Inf. Retr. J.*, 23(3):318–386.
- Enrique Amigó and Stefano Mizzaro. 2020. On the nature of information access evaluation metrics: a unifying framework. *Information Retrieval Journal*, 23(3):318–386.
- Enrique Amigó, Damiano Spina, and Jorge Carrillo-de Albornoz. 2018. [An axiomatic analysis of diversity evaluation metrics: Introducing the rank-biased utility metric](#). In *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*, SIGIR ’18, page 625–634, New York, NY, USA. Association for Computing Machinery.
- Enrique Amigó, Julio Gonzalo, and Stefano Mizzaro. 2023. [What is my problem? identifying formal tasks and metrics in data mining on the basis of measurement theory](#). *IEEE Transactions on Knowledge and Data Engineering*, 35(2):2147–2157.

A. Aspectos técnicos relevantes en el desarrollo de proyectos software

En este apéndice se muestran los aspectos técnicos y de documentación relevantes en el proceso de desarrollo de código. Aunque este aspecto queda, en cierto modo, fuera de los ámbitos del convenio, se ha intentado abordar, hasta donde llegan nuestras posibilidades como universidad, todos los puntos sugeridos.

A.1. Mantenibilidad

Durante el proyecto se han seguido los principios básicos de buenas prácticas de programación, adaptadas en cada caso a su lenguaje de programación: Python y PHP. Entre otros, se han tenido en cuenta las convenciones de nombres de variables, clases, formatos de código, etc., así como estilo de comentarios, descripciones de los métodos, etc. Así mismo, se ha creado un repositorio de documentación en GitHub privado donde se ha ido documentando, en su mayor medida, todos los aspectos técnicos de la implementación de las aplicaciones ODESIA: su arquitectura, herramientas utilizadas en el desarrollo con versiones y dependencias, forma de despliegue, etc.

Por último, se ha habilitado un repositorio GitHub privado para el desarrollo de los proyectos de código, lo que permite el trabajo colaborativo de una forma segura y eficaz, a la vez que permite tener copias de seguridad en caso de posibles fallos.

Por último, y aunque todavía no en pleno desarrollo debido a problemas técnicos en nuestra universidad, se han montado dos servidores, uno para desarrollo y otro para producción. Esta diferenciación, que esperamos poder tener lista próximamente nos permitirá evitar errores de estabilidad en las aplicaciones finales, a la vez que agilizar los procesos de despliegue.

A.2. ENS: Esquema Nacional de Seguridad

La UNED, como centro público de enseñanza, está trabajando desde hace tiempo en certificación de su esquema ENS. Tras una reunión con los responsables de ciberseguridad de nuestra universidad, nos indicaron que actualmente existe un borrador y se está a la espera de próxima aprobación. Por otro lado, los responsables nos indicaron que la infraestructura técnica de seguridad en la UNED supera los estándares y requisitos del esquema ENS. En este contexto, y dado que el desarrollo de este proyecto y el despliegue de las aplicaciones se realiza dentro de la infraestructura de la UNED, todas las aplicaciones ODESIA se encuentran sobre el paraguas de seguridad desarrollado por el servicio de ciberseguridad de la UNED.

Además, se han solicitado los certificados de seguridad, que penderán del certificado de seguridad raíz de la UNED, para poder incluir el protocolo https en las aplicaciones, y que esperamos tener próximamente. No obstante, todas las aplicaciones se ejecutan en una red virtualizada privada, a la que solo tienen acceso los contenedores que así se hayan configurados, con un único contenedor con acceso a Internet. Además, este último contenedor se encuentra protegido por el firewall UNED que da cobertura y seguridad a todos los servidores de la Universidad.

A.3. ENI: Esquema Nacional de Interoperabilidad

La misión y objetivos del Esquema Nacional de Interoperabilidad queda un poco fuera del alcance de este proyecto, dadas las dimensiones del mismo y público objetivo. No obstante, todas las aplicaciones e infraestructuras se están desarrollando teniendo en cuenta que serán usadas en entornos de escritorio y con el navegador más popular: Chrome. Dicho esto, se está haciendo un esfuerzo, dentro de nuestras posibilidades, en adaptar todo lo posible las aplicaciones a otros entornos como móviles, tabletas, etc. Así mismo, se ha utilizado la especificación independiente Swagger para la comunicación entre elementos del proyecto ODESIA para conseguir un mejor control y mantenimiento de los mismos.

A.4. Reglamento General de Protección de Datos

Al igual que en el ENS, el proyecto ODESIA se encuentra bajo el amparo del esquema de protección de datos de la UNED.

A.5. Informe de técnicas de Search Engine Optimization

Las aplicaciones ODESIA son aplicaciones cuyo objetivo no es el posicionamiento en los buscadores, su uso por expertos cualificados, por lo que este punto no ha sido tenido en cuenta dentro del proyecto. No obstante, durante el desarrollo web de las aplicaciones se han usado los estándares de HTML5 con las mejores práctica de desarrollo web dentro de los lenguajes de programación y frameworks.

A.6. Diseño de la navegabilidad

El desarrollo de las aplicaciones ODESIA se han centrado desde sus inicios en el desarrollo de aplicaciones amigables y fáciles de usar. Es por ello que la navegabilidad de las mismas está entre sus objetivos principales, proporcionando todas su funcionalidad mediante como mucho tres clicks.