COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍA DE TELECOMUNICACIONES

TRABAJO FIN DE GRADO

DETECCIÓN Y TRATAMIENTO DE DISLEXIA APLICANDO MODELOS DE MACHINE LEARNING

Autor: Blanca María de Pedro Morejón

Director: Atilano Fernández-Pacheco Sánchez-Migallón

Madrid - Mayo 2023

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Detección y tratamiento de dislexia aplicando modelos de Machine Learning

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2022/2023 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada

de otros documentos está debidamente referenciada.

Blanca.

Fdo.: Blanca María de Pedro Morejón Fecha: 01/ 06/ 2023

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

71216314B ATILANO RAMIRO FERNÁNDEZ-PACHECO

Firmado digitalmente por 71216314B ATILANO RAMIRO FERNÁNDEZ-PACHECO
Fecha: 2023.06.02 07:52:57 +02'00'

Fdo.: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Fecha: 01/06/2023

COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍA DE TELECOMUNICACIONES

TRABAJO FIN DE GRADO

DETECCIÓN Y TRATAMIENTO DE DISLEXIA APLICANDO MODELOS DE MACHINE LEARNING

Autor: Blanca María de Pedro Morejón

Director: Atilano Fernández-Pacheco Sánchez-Migallón

Madrid - Mayo 2023

Agradecimientos

Quiero agradecer a mi familia, en especial a mis padres y a mi hermana, por confiar en mi, apoyarme en todo momento y estar siempre a mi lado. Sin ellos no hubiera sido capaz de tener la determinación y la fortaleza necesaria para enfrentar los desafíos y alcanzar mis objetivos.

A mi abuela Josefina, por quererme de manera incondicional y estar junto a mi en todo momento. Su sabiduría, sinceridad y paciencia me han dado fuerzas para superar cualquier obstáculo que se presente.

A mis amigos, que han estado apoyándome y animándome siempre que lo he necesitado, incondicionalmente.

Por último a mi director de TFG, Atilano, por guiarme, motivarme y permitirme desarrollar el proyecto que quería realizar. Su dedicación, conocimiento y orientación han sido fundamentales en el éxito de mi Trabajo de Fin de Grado.

DETECCIÓN Y TRATAMIENTO DE DISLEXIA APLICANDO MODELOS DE MACHINE LEARNING

Autor: Blanca María de Pedro Morejón
Director: Fernández-Pacheco Sánchez-Migallón, Atilano Ramiro
Entidad Colaboradora: ICAI - Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

El proyecto desarrollado consiste en el diseño de un modelo de Machine Learning que, mediante algoritmos de clasificación, es capaz de detectar si una persona es disléxica o no. Este proyecto también incluye el desarrollo de una aplicación web que incorpora la prueba en la que se basa el conjunto de datos empleado para entrenar el modelo. El usuario tiene a su disposición el test para determinar si padece del mencionado trastorno de aprendizaje, así como información relevante sobre el tema y ejercicios de entrenamiento que pueden realizar de manera dinámica e interactiva para tratar e intentar corregir este trastorno de aprendizaje. La aplicación web se ha construido usando Django y se ha diseñado para ser accesible a todo el público.

Palabras clave: Machine Learning, Accuracy, Detección de dislexia, Accesibilidad

1. Introducción

Machine Learning (ML) es una de las tecnologías más prometedoras en el campo de la salud. En la actualidad, se está implementando en una amplia variedad de ámbitos, desde la mejora en la precisión y eficiencia del diagnóstico hasta el perfeccionamiento de la experiencia y atención que recibe el paciente durante todo el proceso [1].

Los algoritmos de Machine Learning están diseñados para analizar grandes volúmenes de datos, permitiendo detectar patrones dentro de cantidades masivas de información de salud. Esto abre la posibilidad de diagnósticos tempranos y precisos, en etapas donde las enfermedades son más fáciles de tratar [2].

En el campo de la neurología y la educación, se está utilizando Machine Learning con el objetivo de detectar y tratar la dislexia. Gracias a la capacidad de los algoritmos de clasificación para analizar patrones de lectura y escritura en los estudiantes, se está logrando identificar signos de dislexia. Al detectar casos de dislexia de forma temprana, se permite una intervención rápida y eficiente, obteniendo resultados más satisfactorios [3].

2. Definición del proyecto

El propósito de este Trabajo Fin de Grado es poner a disponibilidad de todo aquel que lo necesite herramientas dedicadas a abordar la dislexia, uno de los trastornos del aprendizaje que más casos de fracaso escolar supone [4]. Combinando la tecnología de aprendizaje automático con los distintos estudios que se han realizado sobre la dislexia, se pretende crear un modelo que sirva para detectar con la mayor precisión posible dicho trastorno, ofreciendo a su vez ejercicios para mejorar las competencias afectadas en un entorno dinámico, diseñado de forma accesible y teniendo presente la experiencia de usuario.

El modelo de Machine Learning se va a basar en un estudio realizado en colegios públicos de España en el año 2020, donde participaron más de 5.000 alumnos de entre 7 y 17 años [5]. Está compuesto por 32 preguntas, las cuales se recrearán en la aplicación web, adaptándolas a una experiencia de usuario adecuada, con el objetivo de que esté a disposición del público.

Se van a emplear y comparar distintos algoritmos de clasificación como: K-Nearest Neighbors, Logistic Regression, Support Vector Machines, Decision Tree y Random Forest. Se analizarán distintas medidas de predicción en cada algoritmo, como la exactitud (accuracy)[1], precisión

[1] Métrica que mide la proporción de predicciones correctas realizadas por un modelo de clasificación en relación con el total de predicciones realizadas.

(precision)[2] y exhaustividad (recall)[3], para determinar que modelo ofrece mejores resultados y, por tanto, se implementará en la aplicación web.

Los objetivos de este proyecto son:

- Diseño de un modelo de Machine Learning, utilizando algoritmos de clasificación, capaz de detectar la dislexia.
- Desarrollo de una aplicación web accesible y con una interfaz de usuario óptima.
- Implementación del test de detección de dislexia, junto a ejercicios de entrenamiento, para detectar y tratar el trastorno de aprendizaje.

3. Descripción del sistema

El sistema desarrollado está compuesto por distintos módulos, independientes entre ellos: modelo de detección de dislexia, test de detección de dislexia, ejercicios de tratamiento y sistema de autenticación. Se pueden diferenciar dos grandes bloques: el diseño, entrenamiento y testeo del modelo de Machine Learning y el desarrollo de la aplicación web.

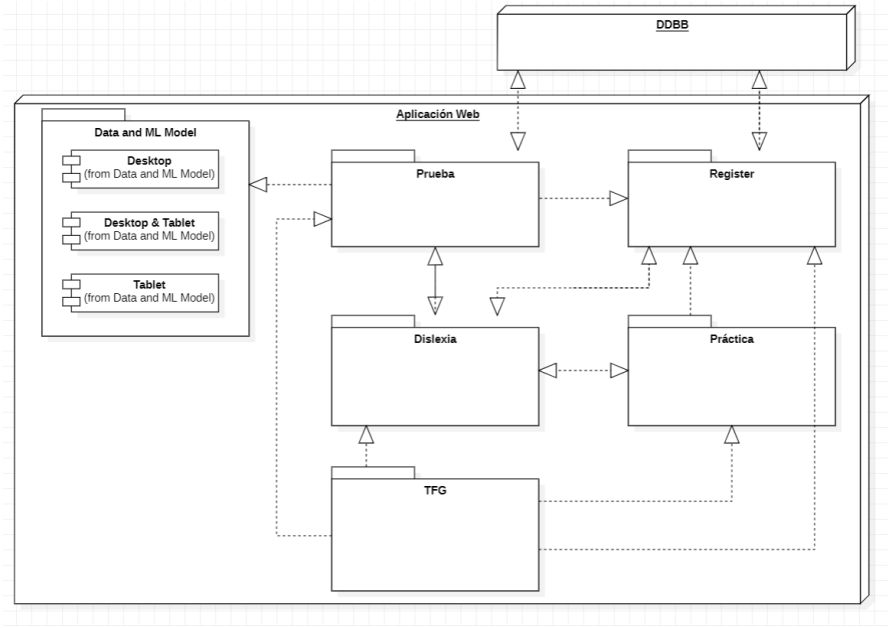


Figura 1: Diagrama de componentes del proyecto

El primero se ha diseñado empleando el lenguaje de programación Python, haciendo uso de las librerías dedicadas a los algoritmos de clasificación, así como las implementadas para manejar la desproporción de clases dentro de un conjunto de datos.

Al haber desarrollado el modelo en Python, el framework de desarrollo web que se ha empleado para la aplicación web es Django. Gracias a su sistema de manejo de datos se ha conseguido gestionar a los usuarios, proporcionándoles funcionalidades como la recuperación de cuentas a través de correo electrónico o la actualización de las contraseñas de forma segura y sencilla.

A su vez, Django ha resultado ser útil y eficiente al contar con un sistema de plantillas y soporte para herencia. Esto ha permitido diseñar una plantilla base de la que han heredado el resto de vistas de cada aplicación[4] y crear interfaces de usuario de manera eficiente y escalable.

[2]Métrica que mide la proporción de verdaderos positivos entre todos los casos clasificados como positivos por un modelo de clasificación.

[3]Métrica que mide la proporción de verdaderos positivos entre todos los casos que realmente son positivos.

[4]Unidades de código que representan una funcionalidad específica del proyecto web.

4. Resultados

Los resultados obtenidos han sido satisfactorios, dado que los objetivos propuestos se han alcanzado satisfactoriamente.

Se ha conseguido diseñar un modelo de Machine Learning capaz de detectar si una persona tiene o no dislexia con una exactitud de un 96 %. Analizando cómo se comportan los distintos algoritmos, junto con las distintas técnicas de gestión de valores nulos y desproporción de clases dentro de un conjunto de datos, se ha conseguido obtener un modelo fiable y con un porcentaje de error mínimo implementando el algoritmo Support Vector Machine:

```
Train set Accuracy:  1.0
Test set Accuracy:  0.944954128440367
Jaccard index 0's:  0.864516129032258
Jaccard index 1's:  0.9151515151515152
```

	precision	recall	f1-score	support
0.0	1.00	0.86	0.93	310
1.0	0.92	1.00	0.96	453
accuracy			0.94	763
macro avg	0.96	0.93	0.94	763
weighted avg	0.95	0.94	0.94	763

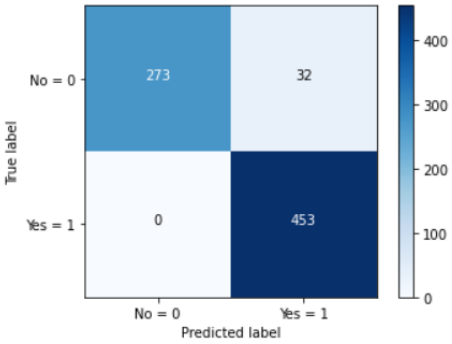


Figura 2: Algoritmo Support Vector Machine

A su vez, se ha diseñado una aplicación web desarrollada sobre Django, compuesta por distintas funcionalidades, como la prueba de detección de dislexia o los ejercicios de entrenamiento, caracterizada por la protección de la información del usuario en la autenticación y un entorno accesible, donde se ha priorizado la experiencia de usuario.

Figura 3: Registro de un usuario nuevo en la web

Finalmente, se han realizado pruebas unitarias y manuales a lo largo del desarrollo de la aplicación web, con el objetivo de asegurar que se cumple con todos los propósitos planteados y que se comporta de forma esperada.

5. Conclusiones

Una vez concluido el desarrollo del proyecto, y en función de los resultados obtenidos, se observa que es viable emplear un modelo de Machine Learning para detectar si una persona tiene dislexia o no, así como el entrenamiento de ciertos procesos cognitivos que se puedan haber visto afectados por dicho trastorno, de forma dinámica y de fácil acceso gracias a la tecnología.

A su vez, se ha demostrado como, con una especificación clara en la etapa de planificación y diseño, se puede diseñar una aplicación web adaptada a todo el mundo, independientemente de las condiciones en las que se encuentre. La web no solo cuenta con un menú personalizable, con el objetivo de que el usuario pueda customizar el texto que se presenta, sino que se ha comprobado que sea compatible con lectores de pantalla y muchos otros criterios marcados por la WCAG[5]

6. Referencias

[1] *How Machine Learning is Revolutionizing the Healthcare Industry*, 2023. [En línea]. Disponible en: `https://www.datasciencecentral.com/how-machine-learning-is-revolutionizing-the-healthcare-industry/`. (Accedido: 09-05-2023).

[2] D. Kriksciuniene. V.s Sakalauskas. Chapter 10 discovering healthcare data patterns by artificial intelligence methods. In *Intelligent Systems for Sustainable Person-Centered Healthcare*, pages 185 – 210. Springer Cham, 2022. (Accedido: 09-05-2023).

[3] P. Raatikainen. J. Hautala. O. Loberg. T. Kärkkäinen. P. Leppänen. P. Nieminen. Machine learning applications in the assessment of speech and language disorders: A scoping review. *Elsevier*, volume 12:pages 7, 2021. [En línea]. Disponible en: `https://www.sciencedirect.com/science/article/pii/S2590005621000345`. (Accedido: 09-05-2023).

[4] S. M. Handler, W. M. Fierson, Section on Ophthalmology, Council on Children with Disabilities, American Academy of Ophthalmology, American Association for Pediatric Ophthalmology, Strabismus, and American Association of Certified Orthoptists. Learning disabilities, dyslexia, and vision. *Pediatrics*, 127(3):e818–e856, 2011. PMID: 21357342.

[5] L. Rello, R. Baeza-Yates, A. Ali, J. P. Bigham, and M. Serra. Predicting risk of dyslexia with an online gamified test. *PLoS One*, 15(12):e0241687, 2020. PMID: 33264301; PMCID: PMC7710040.

[5] https://www.w3.org/TR/WCAG22/

DETECTION AND TREATMENT OF DYSLEXIA APPLYING MACHINE LEARNING MODELS

Author: Blanca María de Pedro Morejón
Director: Fernández-Pacheco Sánchez-Migallón, Atilano Ramiro
Collaborating Entity: ICAI - Universidad Pontificia Comillas

ABSTRACT

The developed project consists of designing a Machine Learning model that, through classification algorithms, is capable of detecting whether a person is dyslexic or not. This project also includes the development of a web application that incorporates the test on which the dataset used to train the model is based. The user has at their disposal the test to determine whether they suffer from the aforementioned learning disorder, as well as relevant information on the topic and training exercises that can be performed in a dynamic and interactive way to treat and attempt to correct this learning disorder. The web application has been built using Django and has been designed to be accessible to all audiences.

Keywords: Machine Learning, Accuracy, Dyslexia detection, Accessibility.

1. Introduction

Machine Learning (ML) is one of the most promising technologies in the field of health. It is currently being implemented in a wide variety of areas, from improving the accuracy and efficiency of diagnosis to perfecting the experience and care received by the patient throughout the process [1].

Machine Learning algorithms are designed to analyze large volumes of data, allowing the detection of patterns within massive amounts of health information. This opens up the possibility of early and accurate diagnoses, at stages where diseases are easier to treat[2].

In the field of neurology and education, Machine Learning is being used with the aim of detecting and treating dyslexia. Thanks to the ability of classification algorithms to analyze reading and writing patterns in students, signs of dyslexia are being identified. By detecting cases of dyslexia early, quick and efficient intervention is allowed, yielding more satisfactory results [3].

2. Project definition

The purpose of this Final Degree Project is to make tools available to anyone who needs them to address dyslexia, one of the learning disorders that leads to the most cases of school failure[4]. By combining machine learning technology with the various studies that have been conducted on dyslexia, the aim is to create a model that can detect this disorder as accurately as possible, while also offering exercises to improve the affected competencies in a dynamic environment, designed to be accessible and with the user experience in mind.

The Machine Learning model will be based on a study carried out in public schools in Spain in 2020, in which more than 5,000 students between the ages of 7 and 17 participated [5]. It consists of 32 questions, which will be recreated in the web application, adapted to an appropriate user experience, with the aim of making it available to the public.

Various classification algorithms, such as K-Nearest Neighbors, Logistic Regression, Support Vector Machines, Decision Tree, and Random Forests, will be used and compared. Different prediction measures in each algorithm, such as accuracy[1], precision[2], and recall[3], will be analyzed to determine which model offers better results and will be implemented in the web application.

[1]Metric that measures the proportion of correct predictions made by a classification model in relation to the total number of predictions made.
[2]Metric that measures the proportion of true positives among all cases classified as positive by a classification model.
[3]Metric that measures the proportion of true positives among all cases that are actually positive.

The objectives of this project are:

- Design of a Machine Learning model, using classification algorithms, capable of detecting dyslexia.
- Development of a web application that is accessible and with an optimal user interface.
- Implementation of the dyslexia detection test and training exercises to detect and treat the learning disorder.

3. System description

The developed system is composed of different modules, independent of each other: dyslexia detection model, dyslexia detection test, treatment exercises, and authentication system. Two large blocks can be differentiated: the design, training, and testing of the Machine Learning model, and the development of the web application.

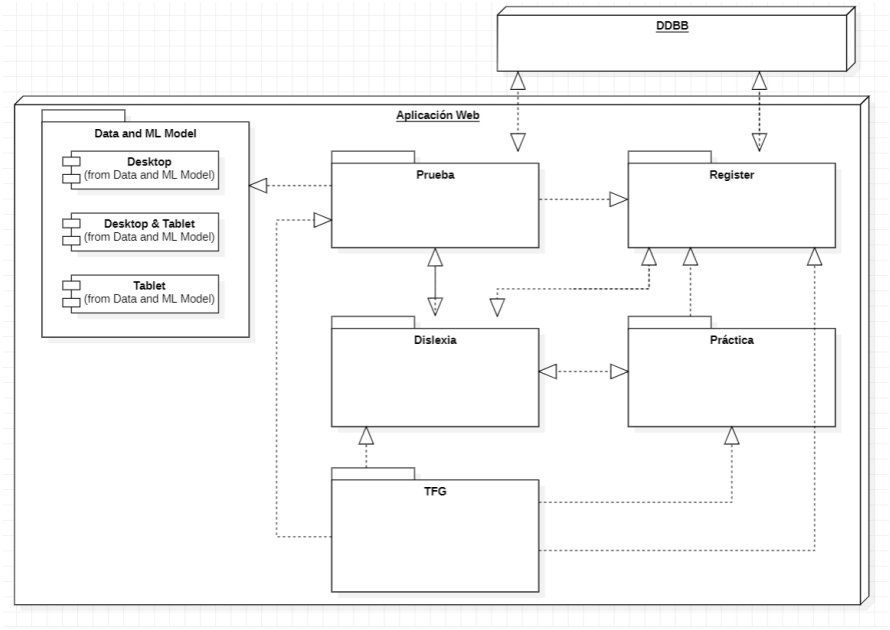


Figura 4: Project component diagram

The first has been designed using the Python programming language, making use of libraries dedicated to classification algorithms, as well as those implemented to handle class imbalance within a data set.

Having developed the model in Python, Django is the web development framework used for the web application . Thanks to its data management system, it has been possible to manage users, providing them with features such as account recovery via email or password updates safely and simply.

In turn, Django has proven to be useful and efficient by having a template system and support for inheritance. This has allowed the design of a base template from which the rest of the views of each application[4] have inherited, creating user interfaces in an efficient and scalable way.

4. Results

The results obtained have been satisfactory, as the proposed objectives have been successfully achieved.

[4]Units of code representing a specific functionality of the web project.

A Machine Learning model has been designed capable of detecting whether a person has dyslexia or not with an accuracy of 96 %. By analyzing how different algorithms behave, along with various techniques for handling null values and class imbalance within a dataset, a reliable model has been obtained with a minimal error rate using the Support Vector Machine algorithm:

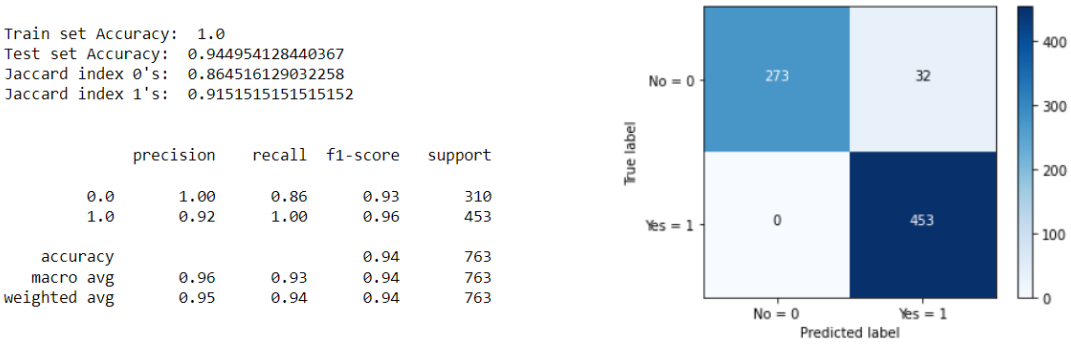


Figura 5: Support Vector Machine Argorithm

What is more, a web application has been designed and developed on Django, composed of various functionalities, such as the dyslexia detection test or training exercises, characterized by user information protection in authentication and an accessible environment, where user experience has been prioritized.

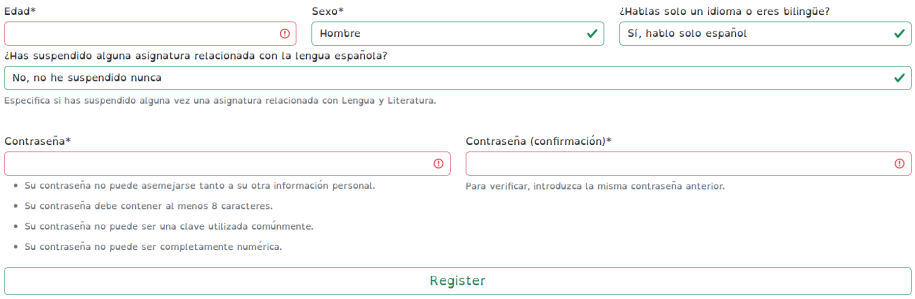


Figura 6: Registro de un usuario nuevo en la web

Finally, unit and manual tests have been carried out throughout the development of the web application, to ensure that all the proposed purposes are met and that it behaves as expected.

5. Conclusiones

Once the project development has concluded, and based on the results obtained, it is observed that it is feasible to use a Machine Learning model to determine whether a person has dyslexia or not, as well as training certain cognitive processes that may have been affected by this disorder, in a dynamic and easily accessible manner thanks to technology.

At the same time, it has been demonstrated how, with a clear specification in the planning and design stage, a web application can be designed to be accessible to everyone, regardless of their conditions. The web not only has a customizable menu, aiming for the user to personalize

the presented text but has also been checked for compatibility with screen readers and many other criteria marked by the WCAG[5].

6. References

[1] *How Machine Learning is Revolutionizing the Healthcare Industry*, 2023. [En línea]. Disponible en: `https://www.datasciencecentral.com/how-machine-learning-is-revolutionizing-the-healthcare-industry/`. (Accedido: 09-05-2023).

[2] D. Kriksciuniene. V.s Sakalauskas. Chapter 10 discovering healthcare data patterns by artificial intelligence methods. In *Intelligent Systems for Sustainable Person-Centered Healthcare*, pages 185 – 210. Springer Cham, 2022. (Accedido: 09-05-2023).

[3] P. Raatikainen. J. Hautala. O. Loberg. T. Kärkkäinen. P. Leppänen. P. Nieminen. Machine learning applications in the assessment of speech and language disorders: A scoping review. *Elsevier*, volume 12:pages 7, 2021. [En línea]. Disponible en: `https://www.sciencedirect.com/science/article/pii/S2590005621000345`. (Accedido: 09-05-2023).

[4] S. M. Handler, W. M. Fierson, Section on Ophthalmology, Council on Children with Disabilities, American Academy of Ophthalmology, American Association for Pediatric Ophthalmology, Strabismus, and American Association of Certified Orthoptists. Learning disabilities, dyslexia, and vision. *Pediatrics*, 127(3):e818–e856, 2011. PMID: 21357342.

[5] L. Rello, R. Baeza-Yates, A. Ali, J. P. Bigham, and M. Serra. Predicting risk of dyslexia with an online gamified test. *PLoS One*, 15(12):e0241687, 2020. PMID: 33264301; PMCID: PMC7710040.

[5]https://www.w3.org/TR/WCAG22/

Índice de la memoria

Índice de figuras

Índice de tablas

Índice de códigos

Capítulo 1

Introducción

La dislexia es uno de los trastornos de aprendizaje más comunes, haciéndose presente en uno de cada diez niños [1]. Se considera una Dificultad Específica de Aprendizaje (DEA), ya que aquellos que la padecen tiene dificultad para reconocer los sonidos del habla, es decir, las vocales, consonantes y sonidos sonoros (mezcla entre vocales y consonantes), lo que desemboca en un inconveniente a la hora de relacionarlos con las letras y las palabras [2].

A inicios del siglo XX, las teorías relativas a la dislexia, originarias de áreas como la psicología cognitiva o las neurociencias, tomaron importancia. A raíz de esto, surgieron gran cantidad de estudios y teorías que profundizaban sobre el concepto de dislexia como discapacidad del aprendizaje [3].

Gracias a estos avances científicos, actualmente se puede clasificar la dislexia en tres categorías diferentes:

1. Según su causa
    - **Del desarrollo**: Trastorno de origen neurológico. Se manifiestan durante el desarrollo infantil.
    - **Adquirida**: Causado a raíz de un trauma o lesión cerebral. Pueden surgir en cualquier etapa de la vida humana.
2. Según la ruta léxica afectada
    - **Fonólogo**: Surge debido a un defecto de la ruta fonológica. Como consecuencia, se presenta dificultad en la correspondencia entre letras y sonidos.
    - **Visual**: Supone un fallo en el funcionamiento en la ruta visual. Presenta inconvenientes al leer palabras irregulares[1].
    - **Mixta o profunda**: Desequilibrio presente en la ruta fonóloga y visual. Manifestación de errores semánticos, junto con los característicos de los otros dos tipos de dislexia (fonólogo y visual).
3. Según su grado
    - **Leve**: Experimenta ciertas limitaciones en las habilidades de aprendizaje.

[1] Palabras no conformes a la normativa de pronunciación de un idioma específico.

- **Moderada**:Experimenta limitaciones significativas en las habilidades de aprendizaje.
- **Grave**: Experimenta limitaciones críticas en las habilidades de aprendizaje.

Ser capaz de clasificar de manera precisa los diferentes tipos de dislexia permite adaptar las necesidades de cada individuo, consiguiendo abordarla y tratarla, con el objetivo de superarla de manera efectiva [4].

Los avances tecnológicos ofrecen una serie de recursos interactivos y dinámicos para el individuo, no solo para superar la dislexia, sino que a su vez pueden usarse como una herramienta para controlar las distracciones del entorno o mantener un seguimiento del proceso realizado, entre otros [5].

Con el auge de la era digital, se han recogido cantidades inmensas de datos de todo tipo. El manejo y análisis de estos volúmenes de datos ha llevado al desarrollo de técnicas avanzadas para su manejo, incluyendo el aprendizaje automático o Machine Learning (ML). Se trata de una rama de la Inteligencia Artificial que utiliza algoritmos y modelos estadísticos para mejorar el rendimiento de sistemas tecnológicos en tareas específicas. El objetivo principal es encontrar y reconocer patrones dentro de estas enormes cantidades de información [6].

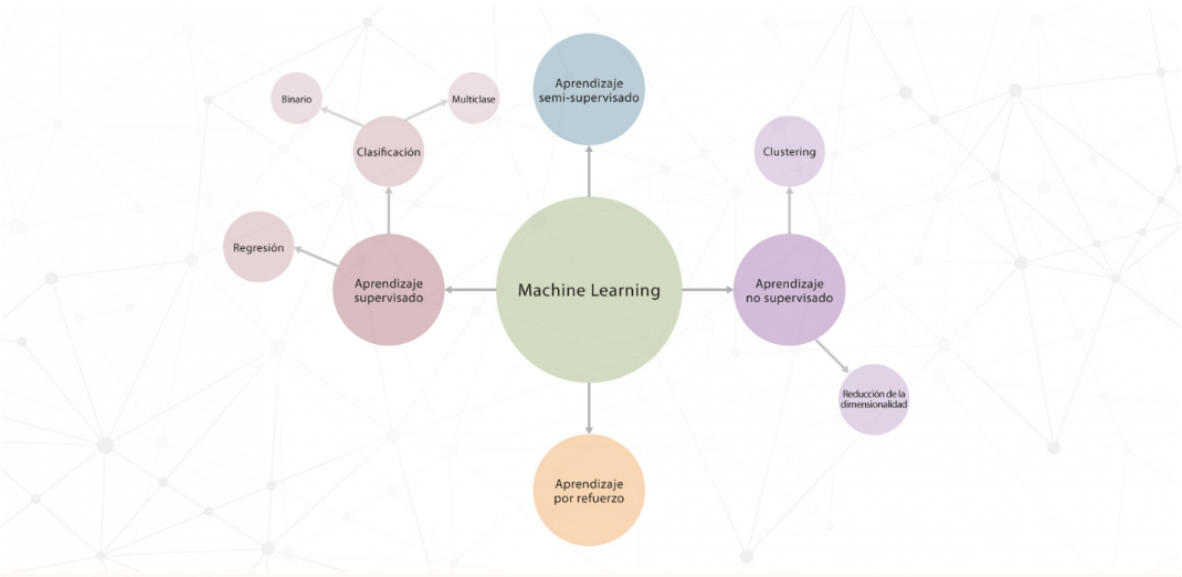


Figura 1.1: Tipos de algoritmos de Machine Learning

Machine Learning puede resultar una herramienta increíblemente útil a la hora de ayudar a detectar este trastorno del aprendizaje. Destaca sobre otras herramientas de predicción y análisis en su capacidad para procesar una gran cantidad de datos, identificando patrones y elaborando predicciones [7].

Los algoritmos de Machine Learning funcionan alimentándose de datos, por ello se dividen en dos o mas grupos de prueba. Los modelos se crean con el objetivo de ser capaces de predecir resultados mediante datos que aún no han sido tratados. Estos dos grupos de datos se pueden diferenciar en uno de entrenamiento (*train set*) para construir el modelo y otro para verificar (*test set*) su correcto

funcionamiento. Además, se pueden utilizar otras herramientas estadísticas para refinar la precisión del modelo [8].

Estas técnicas se están aplicando en varios sectores de la salud para mejorar los diagnósticos, especialmente en las etapas iniciales [9]. Al combinar años de experiencia en el ámbito de la dislexia con la potencia computacional del Machine Learning, los investigadores están desarrollando cada vez más modelos de detección capaces de identificar enfermedades con mayor rapidez y precisión, predecir resultados de pacientes y personalizar tratamientos [10].

Gracias a Internet, el acceso a la información se ha extendido de forma descomunal y ha permitido la creación y el uso de herramientas basadas en Machine Learning a una escala nunca antes vista [11]. Además, la web ha facilitado el desarrollo y despliegue de aplicaciones de Machine Learning que pueden ser utilizadas por cualquier persona con acceso a Internet. Esto incluye aplicaciones diseñadas para el diagnóstico y tratamiento de la dislexia.

De esta forma, se pueden analizar las interacciones del usuario con ejercicios y pruebas específicas para identificar patrones que indiquen la presencia de dislexia [12]. Estos datos podrían ser procesados por un algoritmo de Machine Learning, el cual podría aprender a identificar signos de dislexia con mayor precisión a medida que se expone a más datos.

Una de las ventajas clave de implementar esta herramienta como una aplicación web es su accesibilidad. Las aplicaciones web pueden ser accedidas desde cualquier lugar con conexión a internet, lo que significa que podría ser utilizada por padres, decentes y profesionales de la salud de todo el mundo. Esto no sólo podría mejorar la detección y el apoyo a las personas con dislexia a nivel global, sino que también podría ayudar a generar conciencia sobre este trastorno.

Además, al ser una herramienta digital, los datos generados por los usuarios podrían ser recopilados y analizados para proporcionar una comprensión más profunda de la dislexia. Esto podría ayudar a los investigadores y profesionales de la salud a desarrollar estrategias de intervención más efectivas y personalizadas, y a entender mejor las variaciones individuales en la dislexia.

Capítulo 2

Descripción de las tecnologías

En este capítulo se presentan las distintas tecnologías empleadas para el desarrollo del proyecto. Se han implementado principalmente los software Pyhton, junto a las diversas librerías que ofrece, y Django, con el objetivo de implementar una aplicación web completa, formada por tres bloques principales:

- Interfaz de usuario
- Procesamiento
- Gestión de datos

A su vez, se han empleado herramientas para verificar que la web cumple con las pautas especificadas en la WCAG[1].

2.1. Python

Python es un lenguaje de programación de alto nivel, interpretado y orientado a objetos. Fue creado por Guido van Rossum y lanzado por primera vez en 1991 [13]. Destaca por su diseño elegante y legibilidad de código, lo que lo convierte en un lenguaje muy popular, debido a su fácil comprensión e implementación [14].

Figura 2.1: Logotipo de Python

[1] https://www.w3.org/WAI/standards-guidelines/wcag/es

Se ha decidido desarrollar el proyecto mediante Python, dado que es un lenguaje de programación fácil de leer y entender, gracias a la implementación de sangrías y espacios en blanco utilizado para definir la estructura y los bloques de código.

Por otro lado, destaca por su versatilidad y puede utilizarse en una amplia gama de escenarios, donde se encuentran el desarrollo de aplicaciones web, análisis de datos y aprendizaje automático, entre muchas otras [15]. La comunidad de Python es muy activa y ofrece una gran cantidad de librerías y frameworks, como Django, Pandas, Imblearn y Sklearn, que facilitan el desarrollo de aplicaciones.

2.1.1. NumPy

NumPy es una librería de Python empleada para el cálculo numérico y la manipulación de matrices y vectores multidimensionales. Proporciona una amplia gama de funciones matemáticas y operaciones de matriz que permiten realizar cálculos eficientes y rápidos en Python.

Figura 2.2: Logotipo de NumPy

NumPy es especialmente útil en el ámbito de la ciencia de datos y Machine Learning, ya que permite realizar operaciones numéricas en grandes conjuntos de datos [16].

2.1.2. Pandas

Pandas es una biblioteca de Python diseñada para el análisis y la manipulación de datos estructurados. Ofrece estructuras de datos eficientes, como DataFrames, para organizar, limpiar y procesar datos de manera óptima [17].

Figura 2.3: Logotipo de Pandas

Gracias a la implementación de la librería de Pandas, se ha podido acceder y manipular la información del conjunto de datos del estudio de detección de dislexia permitiendo diseñar el modelo

de Machine Learning posteriormente, ya que ofrece una gran variedad de funciones como la carga de datos desde diferentes fuentes, la manipulación y transformación de datos, el filtrado y la selección de subconjuntos de datos o el manejo de valores nulos y anómalos, entre otras funcionalidades.

2.1.3. Scikit-Learn

Scikit-learn es una librería de Machine Learning de código abierto para Python. Proporciona una amplia gama de algoritmos y herramientas para el aprendizaje supervisado y no supervisado, incluyendo clasificación, regresión, clustering y más.

Figura 2.4: Logotipo de Scikit Learn

Scikit-learn está diseñado para ser fácil de usar y se basa en NumPy, SciPy y Matplotlib para el procesamiento y la visualización de datos [18].

2.1.4. Imbalanced-Learn

Imbalanced-learn es una biblioteca de Python que se centra en el manejo clases desequilibradas dentro de en el ámbito de Machine Learning.

Figura 2.5: Logotipo de Imbalanced Learn

En muchas tareas de clasificación, los conjuntos de datos pueden tener una distribución desigual entre las diferentes clases, lo que puede afectar negativamente el rendimiento y resultados del modelo. Imbalanced-learn proporciona una serie de técnicas de muestreo y métodos de generación de muestras sintéticas para tratar el desequilibrio de clases, como el submuestreo, el sobremuestreo y combinaciones de ambos [19].

2.2. Django

Django es un framework de desarrollo web de alto nivel, escrito en Python, que permite crear aplicaciones web de manera rápida y eficiente. Proporciona un conjunto de herramientas y funcionalidades predefinidas que simplifican el proceso de desarrollo, fomentando la reutilización de código y siguiendo el principio de *baterías incluidas*[2] [20].

Django permite desarrollar numerosas aplicaciones web gracias a las características que incluye por defecto, como la seguridad, la herencia o la relación entre archivos.

Figura 2.6: Logotipo de Django

Incluye una amplia gama de medidas de seguridad integradas con el objetivo de proteger las aplicaciones web e información sensible que puedan manejar. Proporciona protección contra ataques comunes como inyecciones SQL o *cross-site request forgery* (CSRF), entre otros [21]. Además, Django facilita la implementación de autenticación de usuarios y control de acceso [22], así como una gestión segura de las contraseñas y validación de usuario [23].

A su vez, Django utiliza el concepto de herencia en su arquitectura, lo cual permite la creación de modelos[3] y vistas[4] base que pueden ser heredados por otros modelos y vistas. Esto facilita la reutilización de código y la implementación de patrones de diseño de forma óptima y eficiente [24] [25].

Django sigue una estructura de directorios muy definida. La carpeta raíz del proyecto contiene los archivos de configuración principales, como `settings.py` (configuración de la aplicación), `urls.py` (definición de las URLs) y `wsgi.py` (interfaz de servidor web). Aparte, un proyecto puede estar compuesto por numerosas aplicaciones, independientes entre ellas, formadas por una serie de ficheros con funcionalidades específicas [26]. Estos archivos se relacionan de manera coherente para que la aplicación funcione correctamente.

[2]Proporcionar un conjunto completo de funcionalidades y librerías estándar preparadas para su uso, facilitando el desarrollo de aplicaciones y evitando tener que trabajar con módulos externos

[3]Representación de una tabla en la base de datos, definida como una clase de Python, que incluye los campos y las relaciones de esa tabla.

[4]Función o clase que procesa una solicitud HTTP y devuelve una respuesta HTTP, manejando la lógica de negocio asociada a una ruta o URL específica

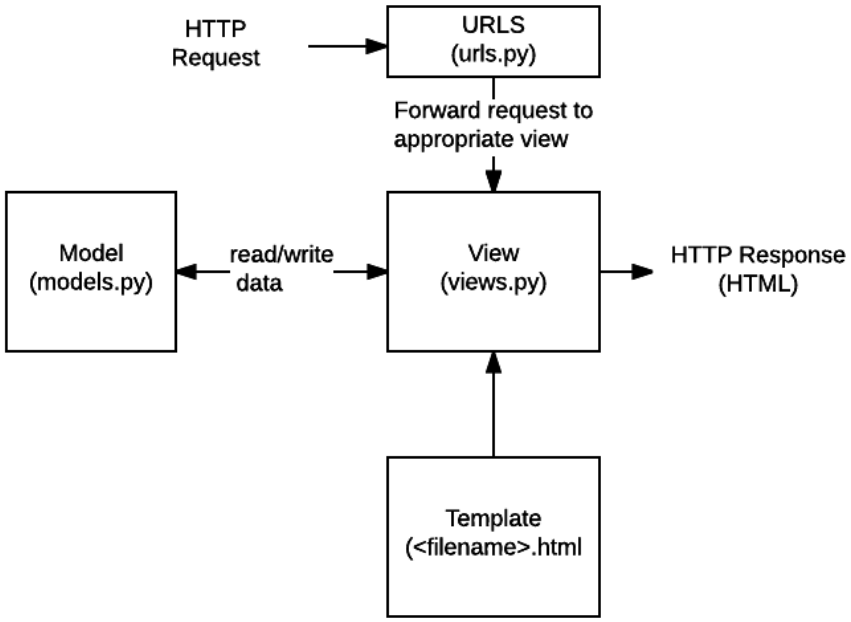


Figura 2.7: Relación de archivos de una aplicación en Django

- Las URLs se emplean para mapear las diferentes páginas y recursos de una aplicación web. El archivo **urls.py** en Django determina qué código se ejecutará en respuesta a una solicitud específica, actuando como el enlace entre las URLs y las vistas correspondientes [27].
- Los modelos en Django representan la estructura y la lógica de la información almacenada en una base de datos. Definen la forma en que se almacenan y se acceden a los datos que maneja la aplicación. Se definen utilizando clases de Python que heredan de la clase **django.db.models.Model** en el archivo **models.py** [24].
- Las vistas en Django son responsables de procesar las solicitudes de los usuarios y generar las respuestas correspondientes. Una vista puede ser una función o una clase que implementa la lógica necesaria para manejar una solicitud. Recibe la solicitud HTTP y puede interactuar con los modelos de la base de datos y tomar decisiones para generar una respuesta, en el archivo **views.py** [25].
- Las plantillas (*templates*) en Django son archivos HTML utilizados para generar la interfaz de usuario de una aplicación web. Pueden contener variables, expresiones condicionales, bucles y otros constructores, permitiendo personalizar la presentación de los datos [28].

2.2.1. SQLite

En cuanto a la gestión de datos, Django ofrece un sistema de ORM (*Object-Relational Mapping*) que permite interactuar con bases de datos de forma intuitiva y sencilla utilizando objetos y consultas en lugar de escribir consultas SQL directamente. Django es compatible con varios motores de base de datos, entre ellos SQLite, que se incluye por defecto [29].

Figura 2.8: Logotipo de SQlite

SQLite es un sistema de gestión de bases de datos relacional ligero y de código abierto. A diferencia de los sistemas de gestión de bases de datos tradicionales, SQLite no requiere un servidor separado y funciona como una biblioteca integrada en la aplicación. Esto supone que la base de datos se almacena en un solo archivo, lo que facilita su portabilidad y distribución [30].

Para utilizar SQLite en Django, se debe configurar la conexión a la base de datos en el archivo **settings.py** del proyecto (directorio raíz). En esta configuración, se especifica la ruta y el nombre del archivo de la base de datos SQLite. Django se encarga de manejar la creación y las migraciones de la base de datos, proporcionando una API (*Application Programming Interface*) de alto nivel para interactuar con la base de datos utilizando modelos y consultas.

SQLite en Django ofrece varias ventajas, como su simplicidad de uso, su bajo consumo de recursos y su portabilidad. Al no requerir un servidor separado, es fácil de configurar y utilizar en el desarrollo y las pruebas locales.

Una de las características por las que destaca la facilidad del manejo de datos en Django es la disposición de una interfaz de administración, la cual ofrece a los usuarios con permisos de administrador un panel de control para gestionar y manipular datos en la aplicación. Cuando un usuario accede a la URL **/admin** de una aplicación Django, se le presenta la interfaz de administración, la que puede manipular los datos almacenados, al permitirle crear, leer, actualizar y eliminar registros de la base de datos [31].

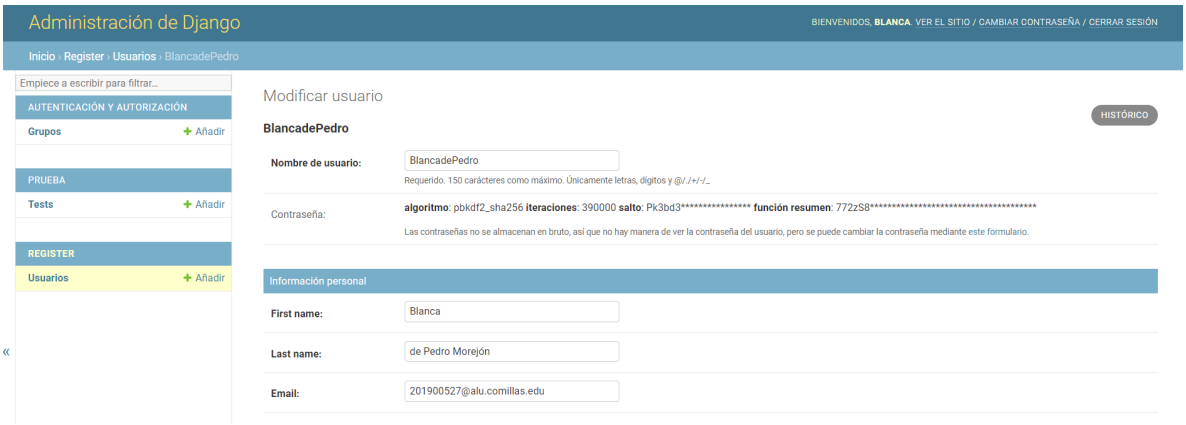


Figura 2.9: Interfaz de administración de Django

La interfaz de administración se puede personalizar modificando la visualización y los campos mostrados, así como permitiendo definir filtros y realizar búsquedas para facilitar la navegación y la gestión de los datos.

2.3. Herramientas de Accesibilidad

Con el objetivo de desarrollar una aplicación web accesible para cualquier tipo de usuario se ha hecho uso, aparte del inspector del navegador web, el cual ha servido para comprobar el correcto etiquetado y la incorporación de un diseño *responsive*, una serie de herramientas para comprobar que las pautas dictadas por la WCAG se cumplen adecuadamente.

La WCAG (*Web Content Accessibility Guidelines*) es un conjunto de pautas desarrolladas por el *World Wide Web Consortium* (W3C) con el objetivo de mejorar la accesibilidad de los sitios web y aplicaciones para personas con discapacidades. El objetivo es establecer recomendaciones técnicas y criterios que los desarrolladores y diseñadores web deben seguir para garantizar que sus contenidos sean accesibles para todos los usuarios, independientemente de su situación personal [32].

2.3.1. NVDA

NVDA (*NonVisual Desktop Access*) es un software de código abierto y gratuito que permite a las personas con discapacidad visual, independientemente del grado que prsenten, acceder al contenido de la web. Funciona como un lector de pantalla que convierte el texto y los elementos visuales en voz o en Braille, lo que facilita la interacción de los usuarios ciegos o con baja visión con aplicaciones y sitios web [33].

Figura 2.10: Logotipo de NVDA

Al utilizar esta herramienta, se puede verificar si una aplicación web está adaptada para personas con discapacidad visual, asegurando que el contenido en su totalidad se pueda obtener a través de lectores de pantalla. Esto garantiza la accesibilidad y la inclusión de las personas con discapacidad visual al proporcionarles una experiencia de usuario óptima.

2.3.2. Colour Contrast Analyser

Colour Contrast Analyser (*Analizador de Contraste de Color*) es una herramienta de accesibilidad que ayuda a evaluar el contraste de color de los elementos textuales o no textuales sobre el fondo en el que se encuentran [34].

Figura 2.11: Logotipo de Colour Contrast Analyser

Esta herramienta permite verificar si los colores utilizados cumplen con las pautas de accesibilidad, especialmente en relación con el contraste entre el texto y el fondo. Es capaz de analizar los valores de contraste según las recomendaciones del W3C[5], lo que ayuda a garantizar que el contenido sea accesible para personas con discapacidad visual o dificultades distinguiendo colores.

2.4. Otras herramientas

Finalmente, en el proceso de diseño, desarrollo y seguimiento del proyecto se han empleado diversas herramientas para facilitar y optimizar el trabajo. Dichas herramientas son las que se muestran a continuación.

2.4.1. Visual Studio Code

Visual Studio Code (*VS Code*) es un editor de código fuente desarrollado por Microsoft. Es multiplataforma y compatible con una amplia gama de lenguajes de programación, entre ellos Python.

Figura 2.12: Logotipo de Visual Studio Code

VS Code ofrece una experiencia de desarrollo completa, contando con características como resaltado de sintaxis, auto-completado, depuración, control de versiones integrado y una amplia gama de

[5]https://www.w3.org/

extensiones que amplían su funcionalidad. Es muy popular en la comunidad del desarrollo web debido a su rendimiento, flexibilidad y comunidad activa [35].

2.4.2. Figma

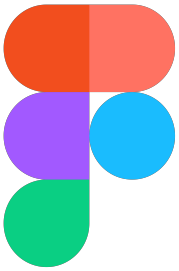


Figura 2.13: Logotipo de Figma

Figma es una herramienta de diseño de interfaces de usuario (UI) y de experiencia de usuario (UX) basada en la nube. Permite a los diseñadores crear diseños de interfaces de usuario, prototipos interactivos y especificaciones de diseño.

Destaca por su capacidad de colaboración en tiempo real y la versatilidad y facilidad con la que se pueden diseñar diversos prototipos. A su vez, ofrece características como bibliotecas de componentes, seguimiento de versiones de diseños y comentarios en línea, lo que agiliza el proceso de diseño y colaboración [36].

2.4.3. Postman

Figura 2.14: Logotipo de Postman

Postman es una herramienta de desarrollo de API que permite probar, documentar y realizar solicitudes HTTP a través de una interfaz amigable y fácil de manejar. Con Postman, los desarrolladores pueden enviar solicitudes a servicios web, realizar pruebas de API, explorar respuestas y realizar automatizaciones [37].

2.4.4. Github

GitHub es una plataforma de desarrollo colaborativo basada en Git[6]. Permite alojar y revisar el código fuente, gestionar proyectos, colaborar con otros desarrolladores, realizar seguimiento de problemas y lanzar versiones de software [38].

[6]https://git-scm.com/

Figura 2.15: Logotipo de Github

En este proyecto se ha utilizado GitHub para alojar el código del sistema desarrollado. El repositorio se puede encontrar en: `https://github.com/BlancadePedro/TFG`.

Capítulo 3

Estado de la cuestión

En los años 30 surgieron los primeros referentes históricos en el campo de la inteligencia artificial, siendo Alan Turing considerado como el padre de la inteligencia artificial. Turing sentó las bases para el desarrollo de la IA con su concepto de la *máquina universal de Turing*[1] y su importante contribución con el *Test de Turing*[2], donde planteaba la posibilidad de que las máquinas pudieran mostrar inteligencia humana [39].

No fue hasta diciembre de 1943, con la publicación del artículo *A logical calculus of the ideas immanent in nervous activity* [40], donde se presentó por primera vez un modelo matemático con las bases para montar una red neuronal.

Sin embargo, el punto de inflexión que experimentó la IA, para llegar a ser como se conoce actualmente, se considera que fue en 1956 en la conferencia *Dartmouth Summer Research Project on Artificial Intelligence*, donde se estableció formalmente el término de Inteligencia Artificial [41]

Una de las ramas que destaca dentro de la Inteligencia Artificial es el Machine Learning (Aprendizaje Automático). El Machine Learning se centra en el desarrollo de algoritmos y modelos de predicción que permiten a los sistemas informáticos aprender automáticamente y mejorar su rendimiento a través de la recopilación y el tratamiento de los datos [6].

En el siguiente capítulo se expondrán las pautas necesarias para desarrollar un modelo de Machine Learning, describiendo como es conveniente abordar el manejo de datos sin limpiar ni tratar, hasta la evaluación correcta del modelo resultante.

Seguidamente, se hablará sobre las diversas tendencias existentes de algoritmos de Machine Learning, analizando cada clase e identificando sus ventajas y defectos.

Posteriormente, se definirán y compararán los distintos métodos que se pueden emplear para la evaluación del modelo diseñado.

[1] Modelo computación capaz de realizar cualquier cálculo de forma automática a una entrada, obteniendo la salida de la misma

[2] Prueba para evaluar la capacidad de una máquina de mostrar un comportamiento inteligente indistinguible del de un ser humano.

Finalmente, se expondrán las soluciones tecnológicas actuales que existen para el tratamiento y la superación de la dislexia.

3.1. Desarrollo de un modelo de Machine Learning

Antes de comenzar con el proceso del desarrollo del modelo de Machine Learning es esencial identificar el problema que se quiere resolver.

Mediante este proyecto se pretende diseñar un modelo de clasificación que, tras haber sido alimentado con un conjunto de datos, sea capaz de identificar si una persona tiene o no dislexia.

A continuación, se muestran las pautas recomendadas para obtener un resultado satisfactorio cuando se quiere diseñar un modelo de Machine Learning desde el inicio [42].

3.1.1. Recopilación, limpieza y preparación de datos

La obtención los datos necesarios para entrenar y evaluar el modelo es un paso fundamental a la hora de desarrollar proyectos de Machine Learning. Implica recopilar datos relevantes y asegurarse de que estén en un formato adecuado.

Además, se deben realizar tareas de preprocesamiento de datos, entre las que se encuentran estandarizar el formato de los datos, manejar valores nulos o gestionar valores anómalos. Es crucial verificar que los datos estén en las condiciones adecuadas para el entrenamiento del modelo, asegurando que los resultados que se obtienen en la etapa final son coherentes y fiables.

3.1.2. Selección de características

El análisis de las distintas características que componen el conjunto de datos es esencial, ya que es necesario comprobar si existen relaciones con otras variables y de que tipo se tratan. Primero se estudia como de relevante es cada variable individualmente y posteriormente el impacto que tiene en el modelo final, con el objetivo de determinar si realmente hay que incluirlas en el modelo final o no.

3.1.3. División del conjunto de datos

Una vez que se tienen los datos preparados, conviene dividir el conjunto de datos inicial en dos subconjuntos, uno dedicado al entrenamiento del modelo, que contendrá la mayoría de la información, y el segundo se empleará en la validación del modelo, comprobando cuantos casos es capaz de predecir correctamente.

El conjunto de entrenamiento se utiliza para entrenar el modelo y el conjunto de prueba se utiliza para evaluar el rendimiento final del modelo.

3.1.4. Selección del algoritmo

Es fundamental seleccionar el algoritmo matemático de Machine Learning que mejor se adapte a la solución que se quiera llevar a cabo. Dependerá fundamentalmente del tipo de problema que se esté abordando (clasificación, regresión, clustering, etc.) y de las características de los datos que

se disponen. Se recomienda explorar diferentes opciones y comparar su rendimiento para tomar una decisión informada y elegir el modelo que mejores resultados proporcione.

3.1.5. Entrenamiento del modelo

Tras haber elegido el algoritmo que mejor se adapta al problema planteado y haber preparado y dividido los datos disponibles, se va a proceder con el entrenamiento del modelo. En esta etapa, el modelo aprenderá a reconocer patrones en los datos y ajustará sus parámetros internos en consecuencia.

Este es un paso puede ser necesario realizar modificaciones con respecto a las decisiones tomadas en los apartados anteriores y ajustar los hiperparámetros del modelo para mejorar su rendimiento y resultados.

3.1.6. Evaluación del modelo

Finalmente, tras entrenar el modelo, se debe evaluar el rendimiento utilizando el conjunto de datos reservado para la evaluación del modelo resultante. Esto permite realizar mejoras adicionales si es necesario. El objetivo es obtener un modelo con un buen rendimiento, para así obtener resultados precisos.

Gracias a este paso se puede estimar como se comporta el modelo ante nuevos datos y si cumple con los objetivos iniciales.

3.2. Algoritmos de Machine Learning

Los algoritmos de Machine Learning son métodos y técnicas utilizados para permitir a los sistemas informáticos aprender automáticamente de los datos y mejorar su rendimiento en tareas específicas sin una programación explícita. Permiten procesar la información con el objetivo de tomar decisiones o predecir el comportamiento de los datos [43].

Como se observó en la Figura 1.1 existen varios tipos de algoritmos de Machine Learning, los cuales se eligen en función de la naturaleza de los datos y el problema que se pretende solucionar. A continuación, se describen los principales tipos de algoritmos de Machine Learning.

3.2.1. Machine Learning supervisado

El Machine Learning supervisado implica el entrenamiento de un modelo utilizando datos etiquetados, es decir, se conoce el valor del resultado. El modelo aprende a mapear las características de los datos de entrada a las etiquetas correspondientes, lo que permite realizar predicciones o clasificaciones en nuevos datos. El modelo tiene la capacidad de .ªprenderüna función matemática que puede determinar la variable objetivo para un conjunto de datos nuevo. [44].

Algunos ejemplos de algoritmos de Machine Learning supervisado incluyen los modelos de regresión, los árboles de decisión y las redes neuronales.

3.2.2. Machine learning sin supervisar

El Machine Learning sin supervisión se utiliza cuando los datos de entrada no están etiquetados y el objetivo es encontrar patrones o estructuras ocultas en los datos. Inicialmente, no se cuenta con información previa sobre valores objetivo o clases, independientemente de si son categóricos o numéricos. Estos algoritmos exploran los datos y buscan similitudes o relaciones, con el objetivo de encontrar patrones [44].

Algunos ejemplos de algoritmos de Machine Learning sin supervisión son el k-means clustering y el agrupamiento jerárquico.

3.2.3. Aprendizaje semisupervisado

El aprendizaje semisupervisado es una combinación de los algoritmos de aprendizaje supervisado y no supervisado. Se cuenta con un conjunto parcialmente etiquetado de datos, es decir, se tiene un número reducido de datos etiquetados y gran cantidad de datos no etiquetados. Se pretende aprovechar tanto los datos etiquetados como los no etiquetados para mejorar la calidad y la eficiencia del modelo [45].

3.2.4. Machine learning por refuerzo

El Machine Learning por refuerzo se basa en un proceso de toma de decisiones en el que un agente aprende a través de la interacción con el entorno. Se caracterizan por no tener "etiquetas de salida", aunque son capaces de aprender y descubrir patrones por sí mismos.

El objetivo es que el agente aprenda a tomar decisiones de forma que se maximicen los resultados obtenidos a lo largo del tiempo. Estos algoritmos son ampliamente utilizados en problemas de toma de decisiones [46].

3.3. Evaluación del Rendimiento de los Algoritmos

La evaluación del rendimiento de los algoritmos de Machine Learning es esencial para medir la eficacia del modelo y determinar si es capaz de devolver los resultados esperados. Hay varias métricas y técnicas que se utilizan para evaluar y comparar los algoritmos de Machine Learning [47].

3.3.1. Matriz de Confusión

La matriz de confusión es una herramienta que muestra la cantidad de predicciones correctas e incorrectas realizadas por un modelo de clasificación en cada clase. Proporciona información sobre los verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, tal y como se muestra en la Tabla 3.1, permitiendo contrastar los resultados obtenidos del modelo.

Proporciona una visión detallada del rendimiento del modelo, en términos de predicciones correctas e incorrectas. Permite obtener una idea genérica, pero completa, sobre el desequilibrio de las clases, así como detectar errores o calcular el valor de diversas métricas, que se expondrán a continuación, con la intención de tomar decisiones en función de los objetivos y las restricciones del problema.

		PREDICTED LABEL	
		Negative	Positive
TRUE LABEL	Negative	True Negative (TN)	False Positive (FP)
	Positive	False Negative (FP)	True Positive (TP)

Cuadro 3.1: Matriz de Confusión

Desde este momento en adelante se hará referencia a los verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos de la que se ha especificado en la Tabla 3.1:

3.3.2. Accuracy

La exactitud (*Accuracy*) es una métrica que calcula la proporción de predicciones correctas realizadas por el modelo en relación con el total de predicciones. Se calcula dividiendo el número de predicciones correctas entre el número total de muestras.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{3.1}$$

3.3.3. Precision

La precisión (*Precision*) mide la proporción de predicciones positivas realizadas por el modelo que verdaderamente son positivas. Resulta útil cuando los falsos positivos resultantes son altos. Se calcula dividiendo el número de verdaderos positivos entre la suma de los verdaderos positivos y los falsos positivos.

$$Precision = \frac{TP}{TN + TP} \tag{3.2}$$

3.3.4. Recall

La sensibilidad o tasa de verdaderos positivos (*Recall*) mide la proporción de casos positivos reales que el modelo es capaz de identificar correctamente. Se calcula dividiendo el número de verdaderos positivos entre la suma de los verdaderos positivos y los falsos negativos.

$$Recall = \frac{TP}{FN + TP} \tag{3.3}$$

3.3.5. F1-Score

El F1-Score es una métrica que combina la precisión y el recall en una única medida. Proporciona una relación equilibrada entre ambas métricas y es útil cuando hay un desequilibrio en las clases del conjunto de datos. Se calcula como la media armónica entre la precisión y el recall.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.4}$$

3.4. Implementación de Machine Learning en el área de la dislexia

Actualmente, existen diversas compañías que se encargan de presentar soluciones tecnológicas para tratar la dislexia. Como se ha mencionado anteriormente, la tecnología puede ser un aliado para las personas con dislexia. Se puede utilizar como una herramienta que facilite los procesos de lectoescritura.

Por ejemplo, mediante un editor de texto específico se puede cambiar la fuente, el tamaño de la fuente, el entrelineado o el espacio del texto, de forma que el contenido sea más accesible para aquellas personas que padezcan de un trastorno de aprendizaje o cualquier tipo de discapacidad.

3.4.1. CHANGE DYSLEXIA

La empresa *CHANGE DYSLEXIA*[3], fundada por Luz Trello, se especializa en la investigación de la dislexia. Su objetivo principal es la reducción global de las tasas de abandono escolar debidas a la dislexia, ya que es la principal razón de fracaso escolar.

CHANGE DYSLEXIA ha creado una aplicación llamada *Dytective*, mediante la cual se puede poner en práctica el tratamiento de la dislexia en un entorno cercano. Pretenden mejorar las habilidades de lectura y escritura en una edad temprana mediante juegos. De esta forma, los más jóvenes se verán atraídos por la diversión que ofrece la aplicación mientras trabajan en habilidades que presentan dificultad para ellos. Está orientado tanto a familias, sanitarios y colegios, con tres planes distintos. Consta más de 40.000 ejercicios, los cuales son capaces de estimular las necesidades y fortalezas de cada niño de forma personalizada, en función de:

- Competencias Lingüísticas
- Memoria de trabajo
- Funciones ejecutivas
- Rendimiento o Desempeño
- Procesos perceptivos

Esta aplicación ya a ayudado a más de 350.000 niños a lo largo del mundo a tratar y superar la dislexia. El objetivo de la fundadora con este proyecto es que la dislexia deje de ser una dificultad oculta y superar las dificultades académicas derivadas de la dislexia así como las barreras socio-económicas mediante becas.

3.4.2. Texthelp

la compañía *Texthelp* [4] se dedica a desarrollar software específico para ayudar a alumnos con dificultades específicas de aprendizaje. Los módulos que ofrecen son:

[3]https://www.changedyslexia.org/
[4]https://www.texthelp.com/en-gb/

- **Read&Write**[5]: herramienta de apoyo de literatura. Ofrece corrección de errores o un diccionario interactivo que para facilitar la lectura.
- **Equatio**[6]: herramienta de apoyo de matemáticas. Permite crear ecuaciones y fórmulas de manera digital.
- **Fluency Tutor**[7]: herramienta para la evaluación de la lectura.
- **Speechstream**[8]: herramienta de apoyo para el aprendizaje literario.

Mediante estas herramientas los niños dispondrán de diccionarios palabra-imagen, es decir, si quieren consultar el significado de algún término dispondrán de una imagen para identificarlo o la opción de hablar al dispositivo móvil y ver como se escribe por pantalla aquello se dicta. A su vez, contiene paquetes relacionados con matemáticas, donde se encuentran herramientas capaces de reconocer los símbolos escritos por cada niño y traducirlos a texto o identificar el contenido de una foto y leerla.

Su objetivo es ayudar y ofrecer apoyo a aquellos niños que tengan dificultades a la hora de leer o escribir, sin hacerles sentir distintos al resto de sus compañeros.

3.4.3. OpenDyslexic

En el área de diseño tipográfico se ha investigado el ámbito de la dislexia. Abelardo González diseñó una fuente específica para personas con dislexia llamada OpenDyslexic[9]. El objetivo detrás del uso de esta fuente es ayudar a leer con mayor facilidad y rapidez, minimizando el número de errores cometidos. Esto se debe al ángulo y grosor que forman las líneas de las letras. Uno de los problemas a lo que se enfrentan las personas disléxicas es la dificultad para diferenciar ciertas letras. Mediante esta fuente, las letras se presentan de forma que sea más fácil para el cerebro identificarlas y que no las rote de manera subconsciente.

Se trata de una fuente libre, lo que implica que está a disposición de aquel que lo necesite. En el desarrollo de esta aplicación se hará uso de la fuente OpenDyslexic, en determinados momentos, con intención de facilitar la experiencia al usuario.

[5]https://www.texthelp.com/en-gb/products/read-and-write-education/
[6]https://www.texthelp.com/en-gb/products/equatio/
[7]https://www.texthelp.com/en-gb/products/fluency-tutor/
[8]https://www.texthelp.com/en-gb/products/speechstream/
[9]https://opendyslexic.org/

Capítulo 4

Definición del Trabajo

Este capítulo tiene como propósito justificar el desarrollo de este proyecto, abordando la motivación detrás del mismo. Se presentarán los objetivos que se pretenden alcanzar, la metodología utilizada para su desarrollo, así como la planificación y estimación económica correspondiente a este proyecto.

4.1. Motivación

El propósito de este proyecto es poner a disponibilidad de todo aquel que lo necesite herramientas dedicadas a superar uno de los trastornos del aprendizaje que más casos de fracaso escolar supone, la dislexia, tal y como se ha expuesto en el Capítulo 1. Combinando la tecnología con toda la información que se ha recopilado sobre la dislexia, se pretende crear un modelo que sirva para detectar con la mayor brevedad posible dicho trastorno, ofreciendo a su vez ejercicios para mejorar las competencias afectadas.

La dislexia no es una enfermedad, por lo que si se diagnostica y trata a una edad temprana las posibilidades de que la vida de un niño no se vea condicionada por el trastorno disminuyen drásticamente. Desgraciadamente, su identificación es muy compleja, ya que los patrones son muy diversos y entre distintos casos no tienen porqué repetirse. Adicionalmente, el sistema escolar no está adaptado para enseñar a niños con dislexia, por ello este trastorno del aprendizaje es responsable del 40 % de los casos de abandono escolar [48].

Por otro lado, aunque sí se haya diagnosticado correctamente la dislexia y exista la posibilidad de que el logopeda [1] pueda intervenir, no todo el mundo tiene acceso a los recursos necesarios para acudir a un procesional sanitario.

Impulsado por la posibilidad de ayudar a través de la tecnología, se ha querido diseñar un test capaz de evaluar aquellos procesos cognitivos comprometidos por la dislexia, como la fonología, morfología, semántica o memoria visual, entre otros, implantando un modelo de Machine Learning que estime, con un rango de error mínimo, si una persona tiene o no dislexia.

[1] Profesional sanitario encargado de tratar las áreas afectadas por alteraciones a nivel de comunicación, lenguaje o habla

A su vez, para aquellos que quieran tratar una habilidad que pueda encontrarse comprometida, como puede ser la memoria o la concentración, se diseñarán una serie de ejercicios interactivos, adaptados a las necesidades de las personas con dislexia, para que puedan practicar aquello que consideran necesario, en un ambiente en el que se sientan cómodos.

La aplicación se ha diseñado priorizando la experiencia de usuario, de forma que sea intuitiva y fácil de navegar. De la misma forma, se ha verificado que la información y las funcionalidades que ofrece sean accesibles para todo el mundo, independientemente de las condiciones en las que se encuentren. Gracias al auge de la tecnología, se ha puesto a disposición de una gran cantidad de usuarios herramientas muy útiles y que pueden impactar positivamente en sus vidas. Sin embargo, no todas están adaptadas al uso de personas con discapacidades. Por ello, se ha tomado como prioridad el desarrollo de una aplicación web accesible.

4.2. Objetivos

Los objetivos principales del proyecto desarrollado son diseñar una página web accesible que permita mejorar la intervención y el tratamiento de la dislexia, ya sea que afecte de forma directa o indirecta. Se implementará como una de las funcionalidades principales un modelo de Machine Learning capaz de predecir, después de su entrenamiento y pruebas, si una persona tiene o no dislexia. De forma más detallada:

- **Modelo de Machine Learning capaz de detectar la dislexia**: se creará un modelo de Machine Learning capaz de estimar, con un margen de error mínimo, si una persona tiene dislexia o no. El diseño del modelo cuenta con dos conjuntos de datos obtenidos del mismo estudio, compuestos por 192 variables, las cuales recogen la información de cada usuario, en base a las cuales se obtendrá el resultado. Se implementarán distintos algoritmos de clasificación, eligiendo el que mejor resultados ofrezca.
- **Diseño de un test interactivo**: con el propósito de identificar si un usuario tiene o no dislexia se va a integrar a la aplicación web un test de 32 preguntas que evalúe distintos procesos cognitivos, como las competencias alfabéticas, fonéticas o léxicas, así como la comprensión y memoria auditiva o visual. El usuario tendrá un tiempo limitado para contestar cada pregunta y en función de las respuestas el modelo de Machine Learning proporcionará un resultado concreto.
- **Diseño de ejercicios para tratar la dislexia**: con la intención de ofrecer una experiencia de usuario completa, se diseñarán una serie de ejercicios que el usuario pueda realizar para tratar distintas habilidades. Se pretende poner a la disposición de todo el público una herramienta interactiva e intuitiva para combatir la dislexia.
- **Aplicación web accesible**: se diseñará una aplicación web que brinde una experiencia de usuario adecuada y sea accesible para todo el público. Se busca crear una página web con un enfoque centrado en el usuario, implementando buenas prácticas como la compatibilidad con diferentes resoluciones de pantalla y un correcto uso de etiquetas en los elementos, facilitando así el uso de herramientas auxiliares y el acceso al contenido de forma simple, independientemente de la situación en la que se encuentre el usuario.

4.3. Metodología

Para el desarrollo del proyecto se va a optar por una metodología ágil, la cual se basa en la producción de software a partir de un enfoque de desarrollo iterativo e incremental. Las tareas a realizar se definirán en distintos *sprints*, que son periodos de tiempo cortos y definidos. En cada sprint se va a establecer y priorizar una serie de tareas a abordar, y se trabajará en ellas de manera cooperativa y flexible.

El objetivo es obtener resultados concretos en cada iteración, permitiendo una mayor flexibilidad y adaptación a medida que se avanza en el proyecto. Esto facilita la retroalimentación constante y la posibilidad de realizar ajustes y mejoras en función de las necesidades y requisitos identificados durante el proceso de desarrollo [49].

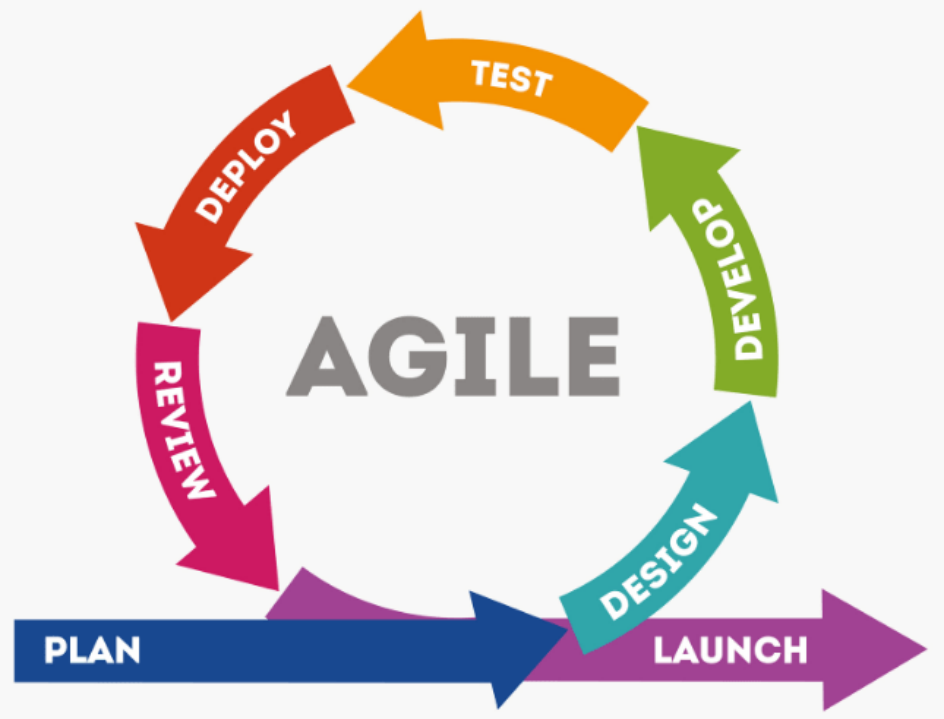


Figura 4.1: Etapas de la Metodología Agile

Cada sprint se focalizará en la resolución de cada una de las funcionalidades que tendrá el proyecto. En paralelo con el desarrollo web, se realizará una documentación detallada, la cual será útil en las etapas de validación y evolución del software.

La razón por la que se ha elegido la metodología ágil es debido a la amplia gama de ventajas que ofrece en comparación con otros enfoques de desarrollo de software. Es conocida por aumentar la

productividad al organizar el trabajo en *sprints*, lo que permite lograr entregas rápidas y frecuentes de código funcional. Esto garantiza que el producto se encuentre en un estado utilizable en todo momento, lo que a su vez permite obtener retroalimentación de los usuarios y realizar ajustes o mejoras [50].

Además, la metodología ágil fomenta la colaboración constante entre los miembros del equipo, promoviendo una toma de decisiones más ágil y efectiva. Estas ventajas combinadas contribuyen a la creación de software de calidad y al cumplimiento de los objetivos del proyecto de manera eficiente.

4.4. Planificación

Se va a mostrar la planificación del proyecto que se ha seguido en la Tabla 4.1 que se muestra a continuación:

	Octubre	Noviembre	Diciembre	Enero	Febrero	Marzo	Abril	Mayo
Familiarización con el Framework Django								
Búsqueda del estudio y conjunto de los datos								
Preparación y limpieza del conjunto de datos								
Diseño del Modelo de Machine Learning								
Testeo y Validación del Modelo Resultante								
Diseño de la Interfaz de la Aplicación Web								
Diseño de la Prueba de Detección de Dislexia								
Incorporación del modelo de Machine Learning								
Diseño de los Ejercicio de Entrenamiento								
Pruebas en la Aplicación Web								
Verificación de la Accesibilidad en la App. Web								
Documentación del Proyecto								

Cuadro 4.1: Planificación del proyecto

El desarrollo de las tareas que se han expuesto en la Tabla 4.1 se han descrito y detallado en el Capítulo 5.

A su vez, se ha hecho uso de la herramienta Trello[2] para la gestión de tareas concretas. Trello permite crear paneles y tarjetas personalizables, lo que ha resultado especialmente útil para dividir las tareas en tres grandes bloques: documentación, diseño del modelo de Machine Learning y desarrollo de la aplicación web.

De esta forma, se pudieron dividir las tareas en tres grandes grupos: tareas por hacer, tareas en proceso y tareas terminadas, permitiendo tener presente en todo momento el progreso del proyecto.

4.5. Estimación económica

Con respecto a la estimación económica del proyecto se han incluido los costes del hardware y software empleados, así como de los trabajadores.

El coste que supone un desarrollador es de 5.000 €/mes. Como se ha observado en la Tabla 4.1 la duración del proyecto ha sido de octubre a mayo, es decir, 9 meses enteros. Esto supone:

$$9 \text{ meses} \times 3.500 \text{ €/mes} = 31.500 \text{ €/desarrollador} \tag{4.1}$$

[2]https://trello.com/es

A su vez, es necesario tener en cuenta los costes de mantenimiento y actualizaciones que requiera la página web anualmente. Se deben incluir reparación de errores, actualizaciones de seguridad y mejoras de rendimiento, así como nuevas incorporaciones y actualizaciones que se incluyan en la aplicación web con el objetivo de crecer, tal y como se expone en el Capítulo 7. Actualmente un desarrollador de software cobra aproximadamente 32.000 €/año[3], por lo que si se hace una estimación de los primeros 5 años el coste del mantenimiento y las actualización del proyecto son:

$$5 \text{ años} \times 32.000 \text{ €/año} = 160.000 \text{ €/desarrollador} \tag{4.2}$$

Los software con los que se ha desarrollado el proyecto han sido gratuitos y de libre acceso, sin embargo si se decide *hostear*, es decir, alojar el proyecto en un servidor web, a nivel empresarial. Se pueden contratar los servicios de AWS, los cuales ofrecen diferentes planes[4], en función del tamaño del tráfico y los recursos que requiere la aplicación web:

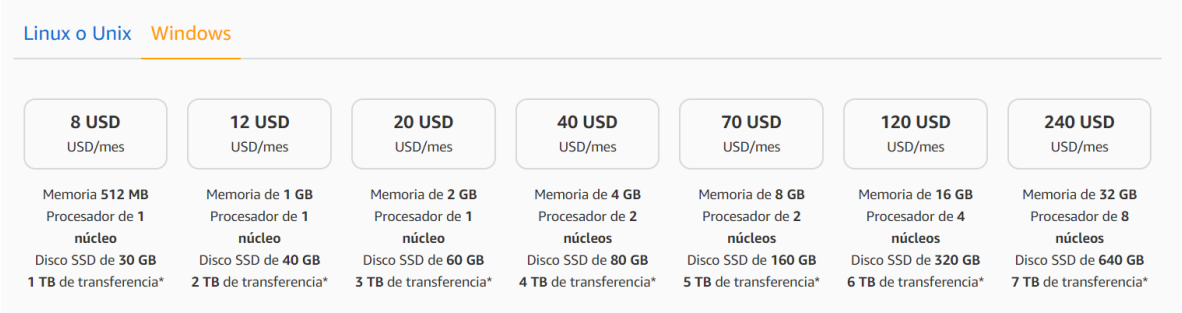


Figura 4.2: Planes disponibles del Servidor de nube virtual de Amazon

Analizando el posible crecimiento que pueda tener el proyecto en cinco años, se pueden establecer tres escenarios posibles. Los costes del desarrollador de software son fijos, sin embargo, el alojamiento en un servidor web y el mantenimiento y la actualización de la aplicación web no:

	Escenario 1	**Escenario 2**	**Escenario 3**
Desarrollador	31.500€	31.500€	31.500€
Mantenimiento y Actualización	32.000€/año	32.000€/año	32.000€/año
Host en la nube	840€/año	1440€/año	2880€/año
Coste Total	195.700€	198.700€	205.900€

Cuadro 4.2: Coste de los Recursos Humanos en los primeros 5 años

A su vez, es necesario tener en consideración los costes del material empleado para el desarrollo del proyecto, los cuales no han sido nulos. Se reflejan en las Tablas 4.3 y 4.4.

[3]https://es.indeed.com/career/desarrollador-de-software/salaries/Madrid–Madrid-provincia
[4]https://aws.amazon.com/es/free/compute/lightsail/

Nombre del dispositivo	DESKTOP-O90KV60
Procesador	Intel(R) Core(TM) i7-1065G7
Memoria RAM	16,0 GB (15,7 GB usable)
Tipo de sistema	Sistema operativo de 64 bits Procesador basado en x64
Sistema Operativo	Windows 11 Home
Pantalla	14". Formato 16:10. IPS WUXGA (1920 x 1200px)
Almacenamiento	460 GB
Precio	1.899,00 €

Cuadro 4.3: Especificaciones del Ordenador Portátil

Tamaño de pantalla	23.8"
Resolución	FHD (1920 x 1080)
Características	Controles en pantalla AMD FreeSync™ Modo Low Blue Light Antirreflectante
Tiempo de respuesta	GtG de 5 ms (con Overdrive)
Conectividad	HDMI; VGA
Precio	126,65€

Cuadro 4.4: Especificaciones de la Pantalla Auxiliar

Por tanto, teniendo en cuenta todos los costes mencionados, el presupuesto final en un periodo de 5 años es:

	Escenario 1	**Escenario 2**	**Escenario 3**
Equipo Hardware	2.025'65€	2.025'65€	2.025'65€
Desarrollador	31.500€	31.500€	31.500€
Mantenimiento y Actualización	32.000€/año	32.000€/año	32.000€/año
Host en la nube	840€/año	1440$€/año	2880€/año
Coste Total	197.725'65€	200.725'65€	207.925'65€

Cuadro 4.5: Coste de la Aplicación en los primeros 5 años

Se ha calculado el coste que supondría mantener el proyecto durante un periodo de 5 años. Ahora se van a calcular los ingresos que se obtendrían si se decidiese vender los servicios que ofrece la aplicación web. Asumiendo que el público objetivo son asociaciones, centros de educación y centros sanitarios, y que lo que se comercializa son subscripciones por usuario o grupo colectivo de usuarios, se van a plantear 3 escenarios con distintos resultados. Antes, Se van a exponer los 3 planes de compra que ofrece la aplicación:

- **Individual: 5€/mes**: Este plan se plantea a familias que quieran tener acceso a la herramienta de forma personal y desde la comodidad de su hogar.

- **Grupo 50 usuarios: 150€/mes**: Este plan está pensado para centros educativos u organizaciones que quieran tener acceso a esta herramienta, consiguiendo ayudar a familias que lo necesiten o colaboradores a un precio más asequible.
- **Grupo 200 usuarios: 400€/mes**: Este plan se orienta a centros sanitarios, concretamente los especializados en trastornos del aprendizaje, quieran incluir herramientas auxiliares que les ayuden con su diagnóstico y entrenamiento.

Cabe destacar que los distintos planes, aunque se han diseñado pensando en un público concreto, pueden ser contratados por cualquier usuario. El objetivo es ofrecer distintas ofertas para que el cliente pueda decidir cual se adapta mejor a sus necesidades.

A continuación, se van a plantear tres escenarios distintos, empezando por el más optimista hasta terminar con el más conservador:

Escenario 1: Optimista

VENTAS	**Precio**	**Suscripciones**	**Total (año)**
Individual	5€/mes	3000	180.000€
Grupo 50 usuarios	150€/mes	200	360.000€
Grupo 200 usuarios	400€/mes	50	240.000€
Ingresos Totales			**780.000€/año**

Cuadro 4.6: Ingresos de la aplicación en el Escenario 1

Haciendo cuentas, en 5 años la cantidad de ingresos asciende a **3,900,000€**, cubriendo los costes mencionados, en los tres escenarios anteriores. Se concluye que la aplicación web es viable y los beneficios son elevados.

Escenario 2: Moderado

VENTAS	**Precio**	**Suscripciones**	**Total (año)**
Individual	5€/mes	2000	120.000€
Grupo 50 usuarios	150€/mes	100	180.000€
Grupo 200 usuarios	400€/mes	25	120.000€
Ingresos Totales			**420.000€/año**

Cuadro 4.7: Ingresos de la aplicación en el Escenario 2

Haciendo cuentas, en 5 años la cantidad de ingresos asciende a **2,100,000€**, cubriendo los costes mencionados, en los tres escenarios anteriores. Se concluye que la aplicación web es viable y los beneficios aunque menores que en el escenario optimista, siguen siendo elevados.

Escenario 3: Conservador

VENTAS	**Precio**	**Suscripciones**	**Total (año)**
Individual	5€/mes	1000	60.000€
Grupo 50 usuarios	150€/mes	50	90.000€
Grupo 200 usuarios	400€/mes	10	48.000€
Ingresos Totales			**198.000€/año**

Cuadro 4.8: Ingresos de la aplicación en el Escenario 3

Haciendo cuentas, en 5 años la cantidad de ingresos asciende a **990,000€**, cubriendo los costes mencionados, en los tres escenarios anteriores. Se concluye que la aplicación web es viable y los beneficios, aunque han disminuido considerablemente con respecto a los dos escenarios primeros, siguen demostrado que es un proyecto rentable. En ningún escenario se pierde dinero, sino al contrario.

A pesar de que los ingresos se han visto disminuidos en el escenario conservador, el total de ingresos aún supera el coste de operación más elevado, asegurando la viabilidad económica del proyecto. Esto es especialmente importante ya que la sostenibilidad financiera a largo plazo es esencial para mantener y desarrollar la aplicación web.

Además, es importante destacar que estos cálculos no incluyen los posibles beneficios adicionales que pueden surgir de la expansión e introducción de nuevas funcionalidades a la aplicación o la colaboración con otras organizaciones. Por lo tanto, este proyecto tiene un potencial considerable para aumentar su rentabilidad y su impacto positivo en la sociedad.

Capítulo 5

Sistema Desarrollado

En este capítulo, se describirá en profundidad el sistema desarrollado. Se empezará exponiendo el proceso de diseño del modelo de Machine Learning para la detección de dislexia a una edad temprana. A su vez, se explicará como se ha realizado la integración entre el modelo de Machine Learning y el test disponible para los usuarios. Se incluirá el procedimiento realizado para el desarrollo de la aplicación web, la cual incluye el cuestionario mencionado y una sección informativa que cuenta con ejercicios prácticos para trabajar algunas de las habilidades que más se ven afectadas a causa de la dislexia.

Con el objetivo de explicar en detalle el sistema creado, se van a diferenciar tres secciones en este capítulo, correspondientes a las distintas fases en las que se va a dividir el proyecto:

1. El desarrollo e implementación del modelo de Machine Learning, asegurando un rendimiento óptimo y la máxima precisión alcanzable. Se evaluarán distintos algoritmos, empleando distintas técnicas para la gestión de valores nulos y la desproporción de clases, eligiendo la opción que mejores resultados ofrezca.

2. El diseño de una aplicación web garantizando una experiencia de usuario apropiada y una aplicación web accesible a todo el público. El objetivo es crear una página web que posea un diseño centrado en el usuario, incluyendo buenas prácticas como la compatibilidad con distintas resoluciones de pantalla o una etiquetación adecuada de los elementos.

3. La integración y verificación adecuadas del flujo entre la prueba de detección de dislexia en la que se basa el modelo de Machine Learning y dicho modelo. En la aplicación web, a su vez, se añadirá una sección informativa, que cuenta con prácticas que pueden realizar los usuarios interesados para trabajar ciertas habilidades que se hayan visto perjudicadas por la dislexia. Se diseñarán y desarrollarán dichos ejercicios.

5.1. Modelo de Machine Learning

Con el objetivo de identificar los casos de dislexia existentes, se llevará a cabo la implementación de distintas técnicas de Machine Learning en un conjunto de datos obtenido tras realizar una prueba compuesta por 32 preguntas a estudiantes de entre 7 y 17 años. El objetivo es identificar patrones que se repitan y en función de los resultados obtenidos diseñar un modelo capaz de predecir si un usuario tiene dislexia o no.

El estudio se realizó en el año 2020, en España, y participaron más de 5.000 personas. Los conjuntos de datos con los que se va a trabajar y sobre los que se va a construir el modelo inicialmente cuentan con 197 columnas compuestas por: el sexo, si su lengua materna es el español o no, si a lo largo de su vida estudiantil se han visto obligados a repetir asignaturas relacionadas con la lengua española o no, la edad, si sufren dislexia o no y los resultados obtenidos de las 32 cuestiones que componen el test. Las características relativas a las cuestiones cuentan con el número de *clicks* que se realiza en cada ejercicio, junto con los *aciertos* y *fallos*, la *puntuación final* de cada cuestión, la *precisión de acierto* y la *tasa de fallo*. Cabe destacar que cada cuestión puede estar compuesta por uno o más ejercicios.

Las pautas que se van a seguir para el desarrollo del modelo de detección de dislexia son:

1. **Recopilación de datos**: Búsqueda de un conjunto de datos que satisfaga los requisitos necesarios para crear un modelo de Machine Learning y permita la creación de una prueba que sea capaz de detectar si un usuario tiene o no dislexia basándose en los resultados obtenidas del estudio.

2. **Limpieza y preparación de datos**: Manejo de valores nulos, estandarización de características y eliminación de outliers. Se analizaran los valores que componen el *dataset* y, con la ayuda de librerías de Python, se modificarán y se adaptarán a las necesidades del modelo, con el fin de obtener el máximo rendimiento.

3. **Selección de características**: Análisis de las distintas características que componen el conjunto de datos, comprobando si existe una fuerte relación con otras variables. Se estudiará como de relevante es cada una individualmente y el impacto que tienen en el modelo final.

4. **Entrenamiento del modelo**: Elección del algoritmo de Machine Learning que mejor se ajuste a los datos disponibles y al objetivo que se quiere cumplir, probando con que parámetros se obtienen mejores resultados. Se diseñará un modelo capaz de detectar con la misma exactitud y precisión los casos positivos como los negativos.

5. **Evaluación del modelo**: verificación de la obtención de resultados apropiados tras el entrenamiento exhaustivo del modelo, valorando su exactitud (*accuracy*[1]), precisión (*precision*[2]) y exhaustividad (*recall*[3]), así como el uso de métricas como el uso de F1-score[4] o las matrices de confusión (Tabla 3.1).

5.1.1. Recopilación de datos

La selección de los datos con que se iba a entrenar el modelo debía cumplir con dos requisitos esenciales: debía ser una muestra representativa de la población, es decir, que el porcentaje de personas con dislexia se encontrase en un rango del 10 % al 15 %, y las variables seleccionadas debían permitir diseñar una prueba accesible al público, que, junto con el modelo de Machine Learning, pudiesen determinar si una persona padece dislexia o no.

Tras una búsqueda exhaustiva se encontraron dos conjuntos de datos procedentes del mismo estudio, que juntos cuentan con más de 5.000 participantes.

[1]Métrica que mide la proporción de predicciones correctas realizadas por un modelo de clasificación en relación con el total de predicciones realizadas

[2]Métrica que mide la proporción de verdaderos positivos entre todos los casos clasificados como positivos por un modelo de clasificación.

[3]Métrica que mide la proporción de verdaderos positivos entre todos los casos que realmente son positivos.

[4]Métrica que relaciona las medidas *precision* y *recall* calculando la media armónica.

Analizando cada cuestión individualmente, se puede observar como se pone a prueba un conjunto de habilidades que afectan directamente a las personas con dislexia, tal y como se muestra en la Tabla 5.1.

Cuestiones	Habilidades puestas a prueba
Cuestión 1 - 4	Conciencia Alfabética y Fonológica Discriminación Visual Categorización
Cuestión 5 - 9	Conciencia Fonológica y Silábica Discriminación Auditiva Categorización
Cuestión 10 - 13	Conciencia Léxica Memoria de Trabajo Auditiva Discriminación Auditiva Categorización
Cuestión 14 - 17	Discriminación Visual Categorización Funciones Ejecutivas
Cuestión 18 - 21	Memoria de Trabajo Visual Auditiva y Secuencial Discriminación Auditiva Categorización
Cuestión 22 - 23	Conciencia Léxica, Fonológica y Ortográfica
Cuestión 24 - 25	Conciencia Morfológica, Semántica y Sintáctica
Cuestión 26 - 27	Conciencia Fonológica, Léxica y Ortográfica
Cuestión 28	Conciencia Silábica, Léxica y Ortográfica
Cuestión 29	Conciencia Fonológica, Léxica y Ortográfica
Cuestión 30	Memoria de Trabajo Visual Secuencial
Cuestión 31	Conciencia Léxica y Ortográfica Memoria de Trabajo Auditiva
Cuestión 32	Memoria de Trabajo Auditiva Secuencial Conciencia Fonológica

Cuadro 5.1: Habilidades tratadas en cada cuestión

Las características de los *dataset* cuentan con la información personal del participante, donde se recoge información como la edad, el sexo o si es español o no, y los resultados obtenidos tras responder

a las 32 cuestiones propias del test.

Cada cuestión consta de seis variables distintas, las cuales se emplearán para determinar como de relevante es el ejercicio el estudio:

- *Clicks*: número de veces que se ha pulsado la pantalla durante el ejercicio.
- *Hits*: número de veces que se ha acertado una pregunta.
- *Misses*: número de veces que se ha fallado una pregunta.
- *Score*: puntuación final sobre una cuestión.
- *Accuracy*: número de aciertos dividido entre el número de clicks.
- *Missrate*: número de fallos dividido entre el número de clicks.

Tras la recopilación y análisis de las características del conjunto de datos elegido, se continuará con el desarrollo del modelo de Machine Learning, avanzando al siguiente punto: la limpieza y preparación de los datos.

5.1.2. Limpieza y preparación de datos

El modelo se ha desarrollado en *Python*, con el apoyo de las librerías *NumPy*, *Pandas* y *Scikit-learn*. Pandas se especializa en el manejo, análisis y procesamiento de datos, y permitirá acceder a la información de los *dataset* y manipular los datos con el objetivo de trabajar exclusivamente con aquellos que sean relevantes para el modelo. Por su parte, Scikit-learn se utilizará para entrenar y evaluar el modelo de aprendizaje automático. Ofrece una amplia gama de algoritmos de Machine Learning, lo que va a permitir probar diferentes modelos y elegir el que mejor se ajuste al conjunto de datos disponibles.

Examinando el conjunto de datos inicial, se identifica que los *dataset* contienen distintos tipos de datos, incluyendo secuencias de texto y valores numéricos de distintos tipos (enteros y flotantes):

	Gender	Nativelang	Otherlang	Age	Clicks1	Hits1	Misses1	Score1	Accuracy1	Missrate1	Clicks2	Hits2	Misses2	Score2	Accuracy2	Missrate2	Clicks3	Hits3	Misses3	Score3	Accuracy3	Missrate3
0	Male	No	Yes	7	10	10	0	10	1.0	0.0	5	5	0	5	1.0	0.0	6	6	0	6	1.0	0.0
1	Female	Yes	Yes	13	12	12	0	12	1.0	0.0	11	11	0	11	1.0	0.0	10	10	0	10	1.0	0.0
2	Female	No	Yes	7	6	6	0	6	1.0	0.0	6	6	0	6	1.0	0.0	6	6	0	6	1.0	0.0
3	Female	No	Yes	7	0	0	0	0	0.0	0.0	0	0	0	0	0.0	0.0	1	1	0	1	1.0	0.0
4	Female	No	Yes	8	4	4	0	4	1.0	0.0	8	8	0	8	1.0	0.0	5	5	0	5	1.0	0.0

Figura 5.1: Datos procedentes de desktop.csv

El conjunto de datos correspondiente a *desktop.csv* se ha designado como *train_df* dado que se va a emplear para entrenar el modelo. Por otro lado, el conjunto de datos correspondiente a *tablet.csv* se ha identificado como *test_df*, puesto que se va a emplear para testear el modelo. Esta nomenclatura se mantendrá a lo largo del modelo. No obstante, cabe destacar que se van a emplear tres combinaciones distintas con los dos *dataset*:

- **Ambos dataset:** *Desktop* para el entrenamiento y *Tablet* para el testeo. No se van a usar técnicas para dividir los dataset ya que cada uno tiene su función establecida.

- **Solo el dataset *Desktop***: Se va a dividir el conjunto de datos de forma aleatoria mediante la función `train_test_split(X, y, test_size=0.2)`. El 80 % de los valores se dedicarán al entrenamiento y el 20 % al testeo.
- **Solo el dataset *Tablet***: Se va a dividir el conjunto de datos de forma aleatoria mediante la función `train_test_split(X, y, test_size=0.2)`. El 80 % de los valores se dedicarán al entrenamiento y el 20 % al testeo.

De esta forma se podrá analizar con que combinación de los *dataset* se obtiene mejores resultados. No obstante, hasta que no se especifique lo contrario, se emplearán ambos *dataset*, uno dedicado al entrenamiento y el otro al testeo del modelo.

Con el objetivo de preparar los datos se sustituirán las secuencias de texto por valores numéricos y se transformarán los valores a flotantes mediante el siguiente código:

```
def prep_Data(ds) :
    #ds.column: access to the column specified
    #map(): itarates through the column specified
    ds['Gender'] = ds.Gender.map({'Male': 0, 'Female': 1})
    ds['Nativelang'] = ds.Nativelang.map({'No': 0, 'Yes': 1})
    ds['Otherlang'] = ds.Otherlang.map({'No': 0, 'Yes': 1})
    ds['Dyslexia'] = ds.Dyslexia.map({'No': 0, 'Yes': 1})

#all numeric
prep_Data(train_ds)
prep_Data(test_ds)

#all float
train_ds = train_ds.apply(lambda x: x.astype('float', errors='ignore'))
test_ds = test_ds.apply(lambda x: x.astype('float', errors='ignore'))
```

Listing 5.1: Conversión a valores numéricos

Se observa como la función `prep_Data(ds)` transforma las columnas con valores textuales a numéricos mediante la instrucción `apply(lambda x : x.astype('float', errors = 'ignore'))`. Se transforman los datos de los dos *dataset* a valores flotantes, lo cual se puede comprobar mediante la función `.dtypes`.

Previo al ajuste de los valores nulos, se va a prescindir de todos los valores mayores a la unidad en las columnas *Accuracy* y *Missrate*. Las variables mencionadas representan porcentajes, por lo que no pueden superar la unidad, sin embargo, si se estudian en detalle se puede comprobar como contienen valores que no cumplen este requisito, pudiendo dar lugar a error si no se corrige. Las dimensiones de los dos *datasets* tras aplicar la eliminanción de los valores mencionados son:

	Training Set
Old Shape	(3644, 191)
New Shape	(2101, 191)

	Testing Set
Old Shape	(1395, 191)
New Shape	(874, 191)

Cuadro 5.2: Dimensiones tras eliminar valores anómalos manualmente

Tras la homogeneización de los datos, se procederá con la gestión de valores nulos [51]. Por ello, se va a comprobar que variables contienen valores distintos a cero recorriendo todas las columnas de cada *dataset* y comprobando si existen valores NaN, añadiendo a una lista la variable indicada si se cumple la condición: `[col for col in dataset.columns if dataset[col].isnull().any()]`.

El conjunto de datos correspondiente a *desktop.csv* no presenta ningún valor nulo. Sin embargo, el conjunto de datos correspondiente a *tablet.csv* sí presenta valores nulos en las variables correspondientes a las cuestiones 13 a la 32. Las técnicas que se van a emplear para el manejo de los valores nulos en el *dataset de testing* son:

- **Eliminación de las filas con valores nulos**: si una fila presenta valores no numéricos, mediante la función `drop()`, se va a prescindir de ella y no se tendrá encuentra en el diseño del modelo.
- **Sustitución por el valor cero**: si se presentan valores no numéricos, se sustituirá el valor NaN por 0 mediante la función `fillna(0)`.
- **Sustitución por la media y la mediana**: si una columna presenta valores no numéricos, se calculará la media y la mediana de dicha columna mediante las funciones `mean()` y `median()`, respectivamente, se sustituirán los valores NaN de cada columna.

El código empleado para llevar a cabo el proceso descrito es:

```
##1 DROP - Drop columns
train_drop = train_df.dropna()
test_drop = test_df.dropna()
##2 ZEROS - Fill with zeros
train_zero = train_df.fillna(0)
test_zero = test_df.fillna(0)
#3.1 MEAN - Mean Fill
train_mean = train_df.fillna(train_df.mean())
test_mean = test_df.fillna(test_df.mean())
#3.2 MEDIAN - Median Fill
train_median = train_df.fillna(train_df.median())
test_median = test_df.fillna(test_df.median())
```

Listing 5.2: Gestión de valores nulos

Al aplicar las técnicas de eliminación de columnas con valores nulos y reemplazo por ceros, los datos que podrían causar problemas se han corregido. Sin embargo, el *dataset* no presenta información sobre las variables de la pregunta 29. La solución adoptada ha sido eliminar las variables relacionadas con la pregunta 29, independientemente de la técnica usada, tanto en el conjunto de datos de entrenamiento como en el de pruebas.

Después de verificar que todos los valores son válidos, se procederá con la eliminación de los valores anómalos. Al haber aplicado cuatro técnicas se tendrán cuatro conjuntos de datos para el entrenamiento y para el testeo. De esta forma, en la etapa de *Evaluación del modelo* se elegirá que técnica escoger en función de la que ofrezca mejores resultados.

5.1.3. Gestión de valores atípicos

Se va a proceder con la eliminación de los *outliers*, es decir, se va a prescindir de aquellos valores que se consideren anómalos o atípicos, puesto que se encuentran fuera del rango esperado.

Con el objetivo de decidir que método se va elegir entre el método Z-score, Rango Intercuartil o Percentil, se va a representar visualmente la distribución que sigue cada variable para identificar la forma en que los datos se comportan, verificando si siguen una distribución normal o no, cómo afecta la presencia de *outliers* y cual es su media y desviación estándar.

A su vez, se va a representar el diagrama de cajas de cada variable, cuyos límites coinciden con el primer y tercer intercuartil, los cuales recogen el 25 % y 75 % de los datos respectivamente, y cuya longitud corresponde al rango intercuartil. La mediana se representa dentro de la caja y la identificación de *outliers* resulta muy sencilla, ya que se consideran anómalos todos aquellos datos que no se encuentran entre:

$$outlier_{inferior} = Q_1 - 1{,}5 \times IQR \tag{5.1}$$

$$outlier_{superior} = Q_3 + 1{,}5 \times IQR \tag{5.2}$$

Debido a que este procedimiento se va a aplicar tanto al conjunto de datos dedicado al entrenamiento como al testeo, diferenciando entre las cuatro técnicas de gestión de valores nulos, se van a mostrar las representaciones gráficas de las variables correspondientes a las cuestiones 1 y 2 [5] con el objetivo de analizar el resultado y razonar que decisión es conveniente tomar:

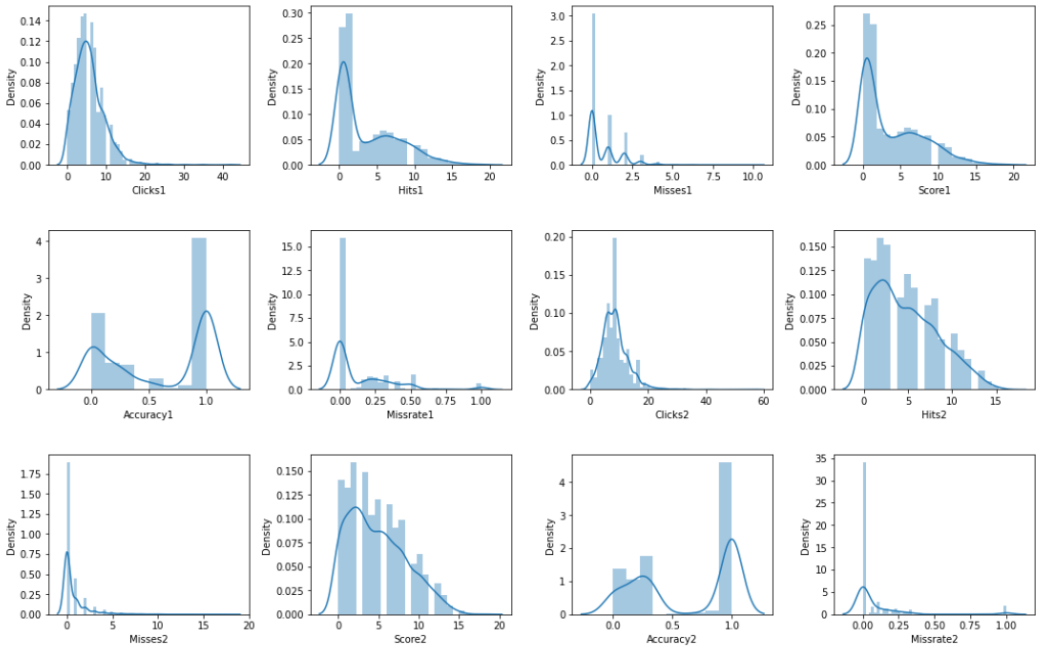


Figura 5.2: Distribución de las cuestiones 1 y 2

[5] La colección completa de las gráfica se mostrará en el Anexo **??**: Distribución gráfica de las variables del modelo de detección de dislexia

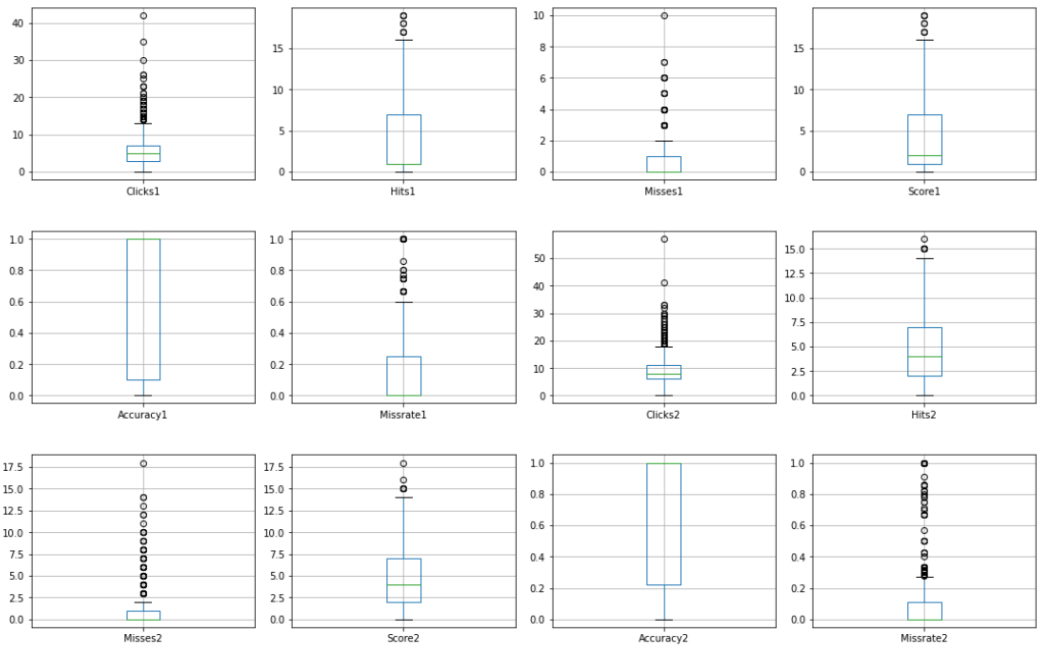


Figura 5.3: Diagrama de cajas de las cuestiones 1 y 2

Tras analizar la distribución de cada variable, y observar como no existe una distribución específica, el método escogido para eliminar aquellos valores distantes del conjunto de datos al que pertenecen es el método del percentil [52]. Se trata de una técnica fácil de aplicar y capaz de adaptarse al *dataset* con el que se trabaja, ya que los límites inferior y superior se ajustan en función de los datos empleados. No requiere de una distribución específica, lo cual supone un punto a favor, dado que no todas las variables tienen la misma distribución, lo que podría suponer un problema utilizando Z-score o el método del Rango Intercuartil [53].

Consiste en un método robusto a la presencia de valores atípicos en los datos, lo que hace que sea menos propenso a verse afectado por ellos, y permite ajustar los límites al *dataset* con el que se está trabajando. En este caso, para no prescindir de un gran número de datos, aun eliminando aquellos que influían negativamente al modelo, se ha establecido como umbral superior el 0.999 e inferior el 0.001:

```
def remove_outliers(ds):
    print("Old Shape: ", ds.shape)
    #access var
    rng = ds.columns
    for col in rng:
        #min threshold
        min_threshold = ds[col].quantile(0.001)
        #max threshold
        max_threshold = ds[col].quantile(0.999)
        #drop outliers
        ds.drop(ds[(ds[col]<min_threshold)|(ds[col]>max_threshold)].index,
            inplace=True)
```

Listing 5.3: Gestión de valores anómalos

Tras limpiar los *datasets*, las dimensiones resultantes son:

	Train
Old Shape	(2101, 186)
New Shape	(1902, 186)

	Test
Old Shape	(236, 186)
New Shape	(50, 186)

Cuadro 5.3: Eliminación - Gestión de valores Nulo

	Train
Old Shape	(2101, 186)
New Shape	(1902, 186)

	Test
Old Shape	(874, 186)
New Shape	(785, 186)

Cuadro 5.4: Sustitución por 0 - Gestión de valores Nulos

	Train
Old Shape	(2101, 186)
New Shape	(1902, 186)

	Test
Old Shape	(874, 186)
New Shape	(774, 186)

Cuadro 5.5: Sustitución por la media - Gestión de valores Nulos

	Train
Old Shape	(2101, 186)
New Shape	(1902, 186)

	Test
Old Shape	(874, 186)
New Shape	(774, 186)

Cuadro 5.6: Sustitución por la mediana - Gestión de valores Nulos

Se puede observar en el Cuadro 5.3 que el *dataset* empleado para testear el modelo, tras eliminar los valores *NaN*, es el más afectado por los valores anómalos. Inicialmente, contenía 236 datos, pero tras la gestión de valores anómalos, su dimensión ha disminuido más de cinco veces. El resto de conjuntos de datos no se han reducido de forma tan drástica, consiguiendo el resultado requerido.

Se va a mostrar cual es la distribución de los datos al inicio de esta sección tras aplicar las técnicas de gestión de valores anómalos, con el objetivo de comparar los cambios:

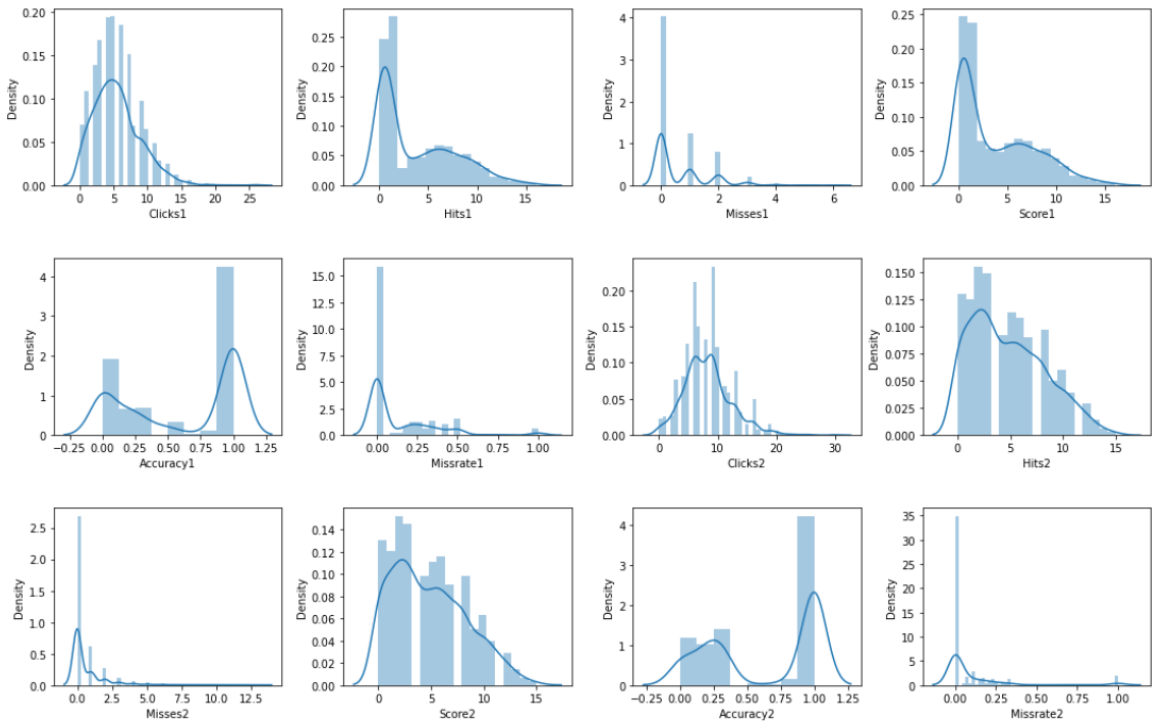


Figura 5.4: Distribución de las cuestiones 1 y 2 tras eliminar los *outliers*

Los límites superiores e inferiores, especialmente los primeros, se han concretado, consiguiendo eliminar aquellos datos que se encontraban demasiado distantes de la mayoría.

Al eliminar los *outliers*, se pretende obtener una visión más precisa del comportamiento de los datos y reducir el impacto que puedan tener en el diseño del modelo, evitando introducir errores de medición o sesgos en el análisis.

Esta idea se ve representada de forma clara en el conjunto empleado para el testeo del modelo, tras sustituir los valores nulos por la mediana de la columna:

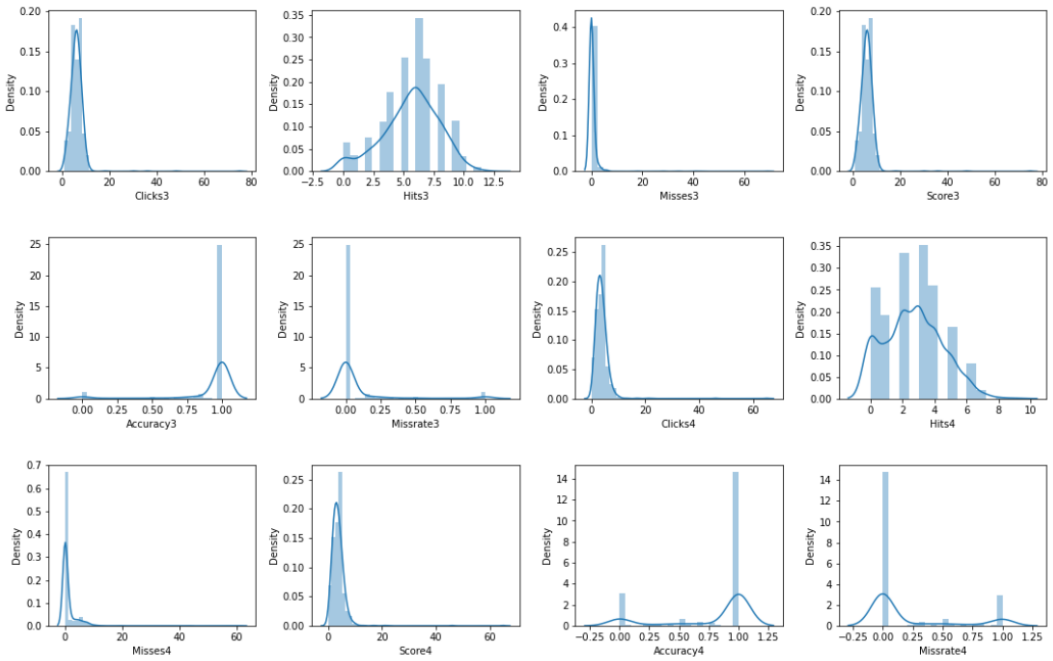


Figura 5.5: Distribución de las cuestiones 3 y 4 antes de eliminar los *outliers*

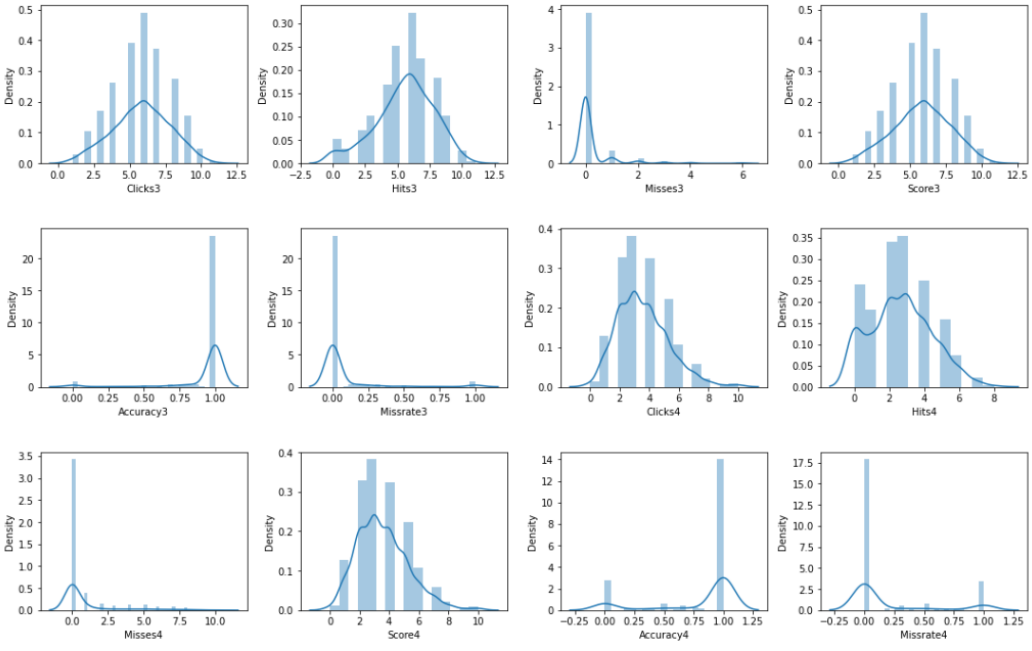


Figura 5.6: Distribución de las cuestiones 3 y 4 tras eliminar los *outliers*

Tras de eliminar los valores atípicos, las variables *Clicks, Hits, Misses y Score* de las cuestiones 3 y 4, no superan el valor 15, lo cual es mucho menor que los resultados iniciales registrados, los cuales llegaban hasta 60 u 80. La forma de la distribución no se ha modificado, aunque actualmente es más fácil observar si las variables siguen una distribución normal.

5.1.4. Selección de características

Se va a llevar a cabo la selección de las características que realmente son relevantes a la hora de diseñar el modelo. Como se ha mencionado al inicio del capítulo, cada cuestión se diseñó evaluando un conjunto de habilidades distintas, las cuales se ven afectadas por la dislexia. Debido a la complejidad que tiene detectar si una persona padece dicho trastorno o no, no se va a eliminar ninguna cuestión, más allá de la pregunta 29 al no contener valores numéricos. Si tras realizar el entrenamiento y el testeo del modelo final se observa que esta decisión puede haber influido negativamente al resultado se tomarán las medidas convenientes.

No obstante, sí se va a evaluar la correlación entre las diferentes variables de cada cuestión. El objetivo es comprobar si existen variables que estén altamente relacionadas entre sí, eliminando una, ya que no aportaría información y se podría simplificar el modelo.

Antes de calcular la correlación entre todas las variables, se va a estudiar cuales pueden tener una mayor relación entre ellas. Cada cuestión tiene seis atributos distintos: *Clicks, Hits, Misses, Score, Accuracy y Missrate*. Analizando la definición de cada una de forma individual se encuentran tres relaciones fácilmente identificables:

$$n^{\circ}\text{Score} = n^{\circ}\text{Hits} \tag{5.3}$$

$$Accuracy = \frac{Hits}{Clicks} \tag{5.4}$$

$$Missrate = \frac{Misses}{Clicks} \tag{5.5}$$

Se puede concluir que el valor de *Score* y *Hits* está directamente relacionado, dado que la puntuación final corresponde al número de aciertos que se haya obtenido en cada ejercicio.

A su vez, los atributos *Accuracy* y *Missrate* están altamente relacionados con *Hits* y *Misses*, respectivamente, ya que su valor se obtiene de los mismos.

Estas conclusiones se han visto respaldadas con el cálculo de las correlaciones indicadas a continuación:

```
def corr_hit_miss(ds, nombre):
    print(nombre+" dataset")
    for i in range(32):
        try:
            print("Question", str(i+1))

            #Pearson correlation coef
            corr_misses = ds['Clicks'+str(i+1)].corr(ds['Misses'+str(i+1)])
            corr_hits = ds['Clicks'+str(i+1)].corr(ds['Hits'+str(i+1)])
            corr_score = ds['Clicks'+str(i+1)].corr(ds['Score'+str(i+1)])
```

```
            #n Hits + n Misses=n Clicks
            ds['Hits+Misses' + str(i+1)]=ds['Hits' + str(i+1)]+ds['Misses' +
                str(i+1)]
            corr_sum=ds['Clicks'+str(i+1)].corr(ds['Hits+Misses'+str(i+1)])

            #show the result
            print("Misses", str(i+1), corr_misses)
            print("Hits", str(i+1), corr_hits)
            print("Score", str(i+1), corr_score)
            print("Hits + Misses", str(i+1), corr_sum)

        except KeyError:
            print("Question "+str(i+1)+" not in the dataset")
```

Listing 5.4: Correlación entre Clicks, Hits, Misses y Score

```
def corr_acc_missrate(ds, nombre):
    print(nombre+" dataset")
    for i in  range(32):
        try:
            print("Question"+str(i+1))

            #Pearson correlation coeff
            corr_acc = ds['Hits'+str(i+1)].corr(ds['Accuracy'+str(i+1)])
            corr_rate = ds['Misses'+str(i+1)].corr(ds['Missrate'+str(i+1)])
            corr_a_r=ds['Accuracy'+str(i+1)].corr(ds['Missrate'+str(i+1)])

            #Hits / Clicks = Accuracy
            ds['Hits/Clicks']=ds['Hits'+str(i+1)]/ds['Clicks'+str(i+1)]
            corr_div_acc = ds['Accuracy'+str(i+1)].corr(ds['Hits/Clicks'])

            #Misses / Clicks = Missrate
            ds['Misses/Clicks']=ds['Misses'+str(i+1)]/ds['Clicks'+str(i+1)]
            corr_div_miss=ds['Missrate'+str(i+1)].corr(ds['Misses/Clicks'])

            #show the result
            print("Accuracy"+str(i+1),corr_acc)
            print("Missrate"+str(i+1),corr_rate)
            print("Between Accuracy and Missrate",corr_a_r)
            print("Accuracy and Hits/Clicks"+str(i+1),corr_div_acc)
            print("Missrate and Misses/Clicks"+str(i+1),corr_div_miss)

        except KeyError:
            print("Question "+str(i+1)+" not in the dataset")
```

Listing 5.5: Correlación entre Clicks, Accuracy y Missrate

Tras ejecutar el código mostrado por pantalla se puede ver como entre *Hits*, *Misses*, *Score* y *Clicks* existe una correlación fuerte, aunque depende de la cuestión que se analice. De forma análoga, se ha estudiado la correlación entre *Accuracy* y *Missrate* y *Hits* y *Misses*, respectivamente, obteniendo una fuerte relación entre ellos.

Tras evaluar los resultados obtenidos, se ha decidido eliminar los atributos *Score*, *Accuracy* y *Missrate*, puesto que su valor se puede obtener a partir de las relaciones mencionadas, por lo que no aportan información nueva. Finalizada esta sección, se va a proceder con el entrenamiento y evaluación del modelo de Machine Learning.

5.1.5. Entrenamiento y Testeo del modelo

El modelo de Machine Learning que se va a diseñar va a ser capaz de predecir si una persona tiene o no dislexia a partir de las variables que se han modificado y preparado en las secciones anteriores. En recuento de casos positivos y de casos negativos inicialmente era:

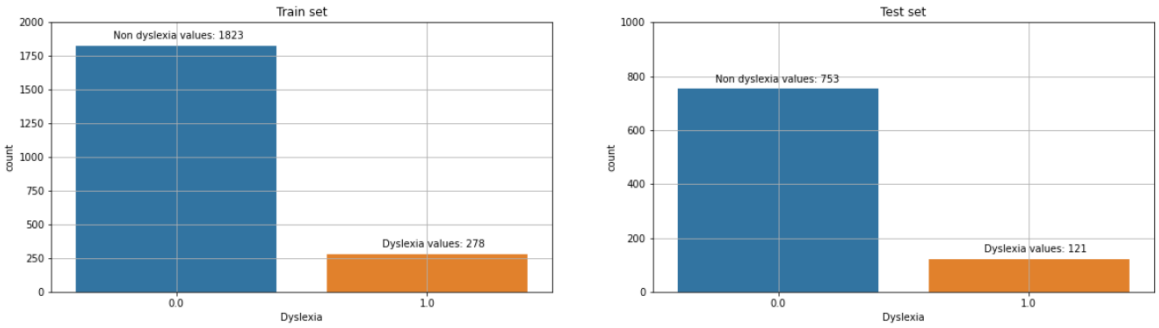


Figura 5.7: Distribución casos positivos y negativos - Dataset originales

Se puede observar como el número de personas que no tienen dislexia (0.0) es mucho mayor en comparación con los que sí tienen dislexia (1.0). Esta diferencia es lógica, ya que la muestra refleja la población real, por lo que solo un 10 % serán casos positivos. Sin embargo, este desequilibrio puede llevar al modelo de Machine Learnig a predecir con mayor precisión los casos que no tienen dislexia, debido a los datos con los que se alimenta. Si se diese el caso, habría que usar técnicas de gestión de desproporción de valores.

Los algoritmos de Machine Learning escogidos para el entrenamiento del modelo son algoritmos de aprendizaje supervisado, es decir, el modelo se va a entrenar mediante datos etiquetados (*Dyslexia*: 0 o 1). Son ideales para los problemas de clasificación, dado que permiten predecir un resultado en función de los datos de entrada. El modelo se va a entrenar con cinco algoritmos diferentes, escogiendo el que mejores resultados ofrezca:

- **K-Nearest Neighbors**: Clasifica objetos basándose en las características de los datos de entrenamiento, es decir, los clasifica en función de las similitudes entre sus características y la de las características de sus vecinos. Para obtener un modelo óptimo, es esencial seleccionar un valor adecuado de K, que representa el número de vecinos más cercanos, ya que un valor incorrecto puede derivar en resultados menos precisos [54].
- **Logistic Regression**: Pretende modelar la probabilidad de que un objeto pertenezca a una clase determinada en función de sus características. Asume una relación lineal entre las características y la probabilidad de la clase objetivo y utiliza una función logística para estimar dicha probabilidad [55].

- **Support Vector Machines**: Pretende encontrar un hiper-plano que maximice la separación entre las clases en el espacio de características. Se trata de un algoritmo convexo, es decir, siempre converge a una solución global óptima [56].
- **Decision Tree**: Construye un árbol de decisiones a partir de los datos de entrenamiento, donde cada nodo del árbol representa una pregunta sobre las características de los datos y cada rama una respuesta a dicha pregunta [57].
- **Random Forest**: Son una extensión de los Árboles de Decisión que combinan múltiples árboles para mejorar la precisión. Construyen múltiples árboles de decisión utilizando diferentes subconjuntos aleatorios de características y combinando las predicciones resultantes, con el objetivo de obtener una predicción final [58].

En la fase de entrenamiento se va a implementar la técnica de validación cruzada *K-fold cross validation*. Introduciendo como argumentos de entrada los datos originales, diferenciando entre el conjunto de datos con las características y el conjunto de datos que contiene los valores a predecir, y el modelo que se quiere testear se va a entrenar el modelo 10 veces (k = 10, valor modificable) dividiendo los datos dedicados al entrenamiento y al testeo de forma distinta en cada iteración. La función devuelve la media entre todas las medidas de precisión calculadas, obteniendo una estimación más precisa y evitando el sobre-ajuste del modelo.

Es importante resaltar que los algoritmos mencionados se van a aplicar a cada técnica de gestión de valores nulos. Es decir, se van a modelar:

$$\text{Modelos} = 5 \text{ Algoritmos} \times 4 \text{ Técnicas Nulos} \times 3 \text{ Combinaciones Datasets} \tag{5.6}$$

El parámetro *Combinaciones Datasets* hace referencia a la combinación de los diferentes conjuntos de datos que ofrece el estudio en el que se está basando este trabajo, tal y como se aclaró en la sección *Recopilación de datos*

A continuación, se van a mostrar los resultados obtenidos tras implementar cada algoritmo mencionado, mediante la matriz de confusión y las medidas de evaluación correspondientes a cada técnica.

K-Nearest Neighbors

La búsqueda por el algoritmo de Machine Learning que mejor prediga si una persona tiene dislexia o no va a comenzar con ***K-Nearest Neighbors***. Se trata de uno de los algoritmos más sencillos de implementar, y que más ha sido empleado para la detección de patrones. Clasifica los objetos en función de las similitudes que presenten con el restos de datos empleados para el entrenamiento, basándose en una medida de distancia, que podría ser *Distancia Euclidea, Manhattan, Minkowski o Hamming*, entre otras.

El algoritmo calcula la distancia entre el valor que se quiere clasificar y los demás valores del conjunto de datos, asignándolo a un grupo en función de la distancia que presente con sus vecinos. Los vecinos se identificarán como los k objetos más cercanos y el valor se asignará al grupo al que pertenezcan la mayoría de sus vecinos, por ello es esencial escoger cuidadosamente el valor K.

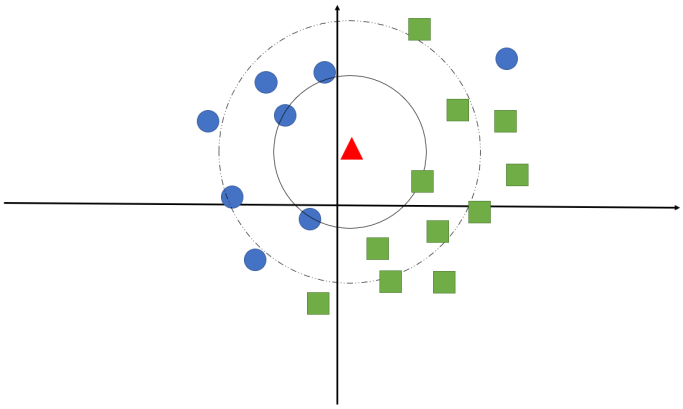


Figura 5.8: Representación algoritmo K-NN

La medida de distancia empleada para el diseño del modelo de Machine Learning ha sido *la distancia Euclídea*. Se ha escogido esta medida dado que es intuitiva y ampliamente utilizada para el cálculo de distancias en los algoritmos de aprendizaje automático. Dicha distancia se calcula:

$$D(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5.7}$$

De esta forma se podrá calcular de forma sencilla la distancia entre los distintos K vecinos y asignar los valores a un grupo determinado.

Aunque clasificar los datos en función de la distancia entre los K vecinos más cercanos es un proceso sencillo e intuitivo de implementar, es necesario tener en cuenta que este algoritmo supone un alto coste computacional y de memoria. Al ser un algoritmo basado en instancias no aprende explícitamente como funciona el modelo que está entrenando, sino que opta por memorizar las instancias de entrenamiento que posteriormente se utilizan como referencia y utilizarlas como referencia en la fase de predicción.

A su vez, es imprescindible escoger rigurosamente el valor de K, parámetro que se elegir en el momento de la construcción del modelo. No existe un número óptimo de vecinos, es necesario elegir el valor de K en función de cada modelo. Por ello, se ha diseñado una función que calcule la medida de precisión (*score accuracy*) empleando distintos K's e imprima por pantalla los resultado obtenidos. De esta forma se podrá valorar que valor de K se considera mejor opción, teniendo en cuenta que cuanto menor sea K, menor sesgo tendrá el modelo aunque contará con una alta varianza.

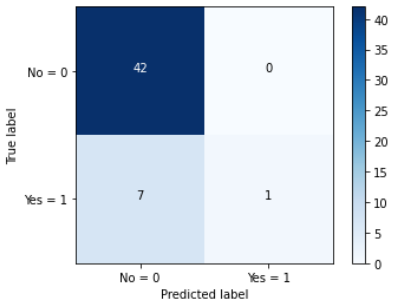
Es importante destacar que, al trabajar con un algoritmo basado en distancias como KNN, se deben estandarizar los datos al inicio del modelo para ayudar al algoritmo a converger más rápidamente, lo que supondrá una reducción en el tiempo de entrenamiento. La estandarización también evita que las variables se encuentren en distintas escalas, lo que podría provocar una desproporción en los pesos de cada una, dando más importancia a ciertas características, cuando todas deben tener el mismo peso.

A continuación, se va a mostrar el resultado obtenido tras entrenar y evaluar el modelo empleando el algoritmo K-Nearest Neighbors a cada técnica de gestión de valores nulos:

```
Train set Accuracy:  0.9043112513144059
Test set Accuracy:  0.86
Jaccard index 0's:  0.8571428571428571
Jaccard index 1's:  0.125
```

	precision	recall	f1-score	support
0.0	0.86	1.00	0.92	42
1.0	1.00	0.12	0.22	8
accuracy			0.86	50
macro avg	0.93	0.56	0.57	50
weighted avg	0.88	0.86	0.81	50

```
<Figure size 432x288 with 0 Axes>
```

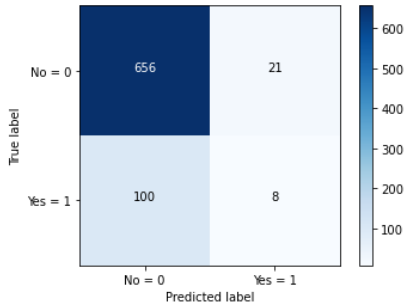


(a) KNN-Drop Technique

```
Train set Accuracy:  0.9043112513144059
Test set Accuracy:  0.845859872611465
Jaccard index 0's:  0.8442728442728443
Jaccard index 1's:  0.06201550387596899
```

	precision	recall	f1-score	support
0.0	0.87	0.97	0.92	677
1.0	0.28	0.07	0.12	108
accuracy			0.85	785
macro avg	0.57	0.52	0.52	785
weighted avg	0.79	0.85	0.81	785

```
<Figure size 432x288 with 0 Axes>
```

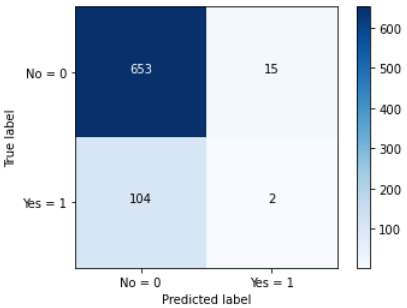


(b) KNN-Zero Technique

```
Train set Accuracy:  0.9043112513144059
Test set Accuracy:  0.8462532299741602
Jaccard index 0's:  0.8458549222797928
Jaccard index 1's:  0.01652892561983471
```

	precision	recall	f1-score	support
0.0	0.86	0.98	0.92	668
1.0	0.12	0.02	0.03	106
accuracy			0.85	774
macro avg	0.49	0.50	0.47	774
weighted avg	0.76	0.85	0.80	774

```
<Figure size 432x288 with 0 Axes>
```

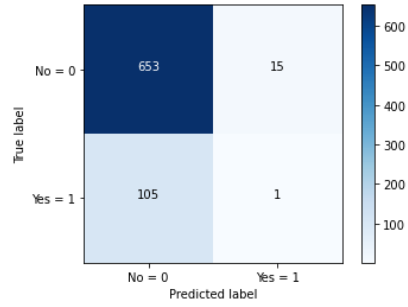


(c) KNN-Mean Technique

```
Train set Accuracy:  0.9043112513144059
Test set Accuracy:  0.8449612403100775
Jaccard index 0's:  0.8447606727037517
Jaccard index 1's:  0.008264462809917356
```

	precision	recall	f1-score	support
0.0	0.86	0.98	0.92	668
1.0	0.06	0.01	0.02	106
accuracy			0.84	774
macro avg	0.46	0.49	0.47	774
weighted avg	0.75	0.84	0.79	774

```
<Figure size 432x288 with 0 Axes>
```



(d) KNN-Median Technique

Figura 5.9: Matriz de confusión de K-Nearest Neighbors

Logistic Regression

El siguiente algoritmo que se va a implementar es***Logistic Regression***. Este algoritmo pertenece a la categoría de Clasificador lineal, dado que es ampliamente empleado para la clasificación de variables binarias, aunque se puede emplear para clasificar problemas multiclase de la misma forma. Se trata de un algoritmo sencillo de implementar y muy intuitivo a la hora de interpretar resultados, dado que se basa en la *función Sigmoide o Logística*, de ahí su nombre.

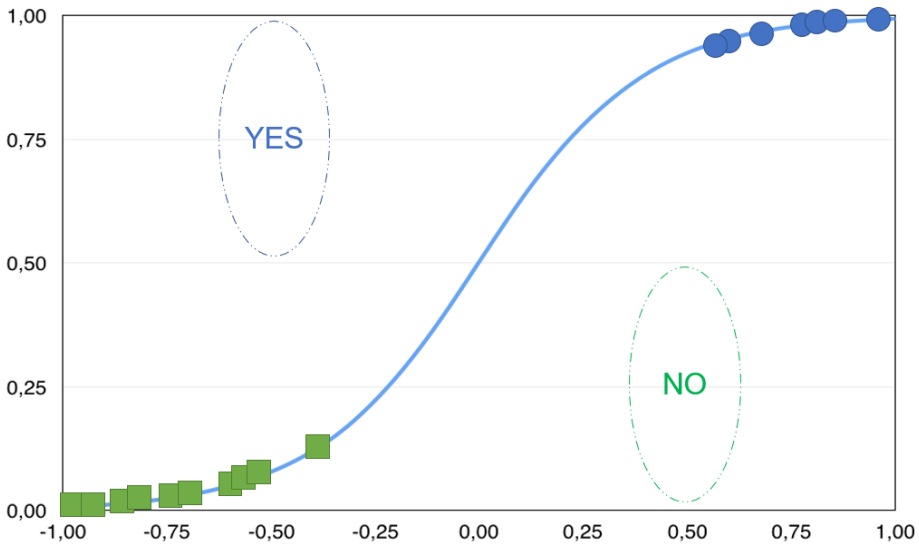


Figura 5.10: Representación algoritmo Logistic Regression

La función logística predice la probabilidad de que una observación pertenezca a una determinada clase. Es decir, siendo z una combinación lineal de la suma de las características de la observación a clasificar multiplicadas por el peso correspondiente, transforma dicho valor en una probabilidad entre 0 y 1, permitiendo clasificar la observación. La expresión matemática que sigue la función Sigmoide es:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{5.8}$$

Donde z:

$$z = w_1 x_1 + w_2 x_2 + ... + w_n x_n + b \tag{5.9}$$

La función logística recibe a su vez el nombre sigmoide, debido a la forma de "S" que adopta la curva. Analizando la expresión matemática se puede observar que si el valor de z es elevado, la probabilidad de pertenecer a la clase positiva se acerca 1. Mientras que si z disminuye, la probabilidad se acerca a 0. De esta forma se puede modelar una relación entre las características del conjunto de datos empleado y la variable de salida:

$$P(Y = 1|z) = 1 - P(Y = 0|z) \tag{5.10}$$

Al implementar *Logistic Regression* en Python se ha tenido que definir el valor del atributo *solver*. Este parámetro se utiliza para optimizar la función de pérdida , es decir, mide el grado en el que se

ajusta el modelo de Machine Learning a los datos empleados para el entrenamiento. Existen numerosos algoritmos que se encargan de la optimización del modelo, como *liblinear, newton-cg, lbfgs o saga*, entre otros.

Se ha escogido *liblinear* dado que los conjuntos de datos que se están empleando para el diseño del modelo no son excesivamente grandes. Emplea un algoritmo de descenso en gradiente para la optimización y es el recomendado cuando los datos no son muy elevados.

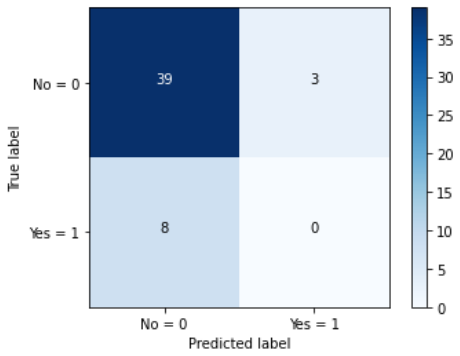
Al igual que en *K-Nearest Neighbors* se van a estandarizar los *dataset* de entrenamiento y testeo, consiguiendo que el algoritmo converja más rápido, todas las características se encuentren en la misma escala y que la interpretación de los coeficientes sea más sencilla.

A continuación, se va a mostrar el resultado obtenido tras entrenar y evaluar el modelo empleando el algoritmo Regresión Logística a cada técnica de gestión de valores nulos:

```
Train set Accuracy:  0.8948475289169295
Test set Accuracy:  0.78
Recall 0's:  0.9285714285714286
Recall 1's:  0.0
F1 score 0's:  0.8764044943820225
F1 score 1's:  0.0
Jaccard index 0's:  0.78
Jaccard index 1's:  0.0
```

	precision	recall	f1-score	support
0.0	0.83	0.93	0.88	42
1.0	0.00	0.00	0.00	8
accuracy			0.78	50
macro avg	0.41	0.46	0.44	50
weighted avg	0.70	0.78	0.74	50

```
<Figure size 432x288 with 0 Axes>
```

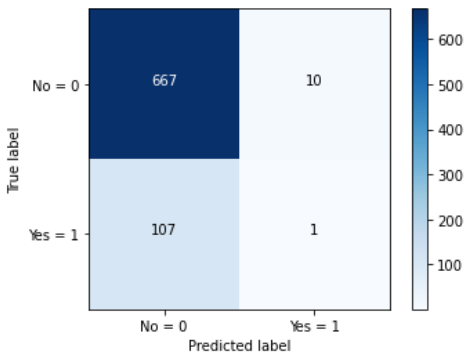


(a) LR-Drop Technique

```
Train set Accuracy:  0.8948475289169295
Test set Accuracy:  0.8509554140127389
Recall 0's:  0.9852289512555391
Recall 1's:  0.009259259259259259
F1 score 0's:  0.9193659545141282
F1 score 1's:  0.01680672268907563
Jaccard index 0's:  0.8507653061224489
Jaccard index 1's:  0.00847457627118644
```

	precision	recall	f1-score	support
0.0	0.86	0.99	0.92	677
1.0	0.09	0.01	0.02	108
accuracy			0.85	785
macro avg	0.48	0.50	0.47	785
weighted avg	0.76	0.85	0.80	785

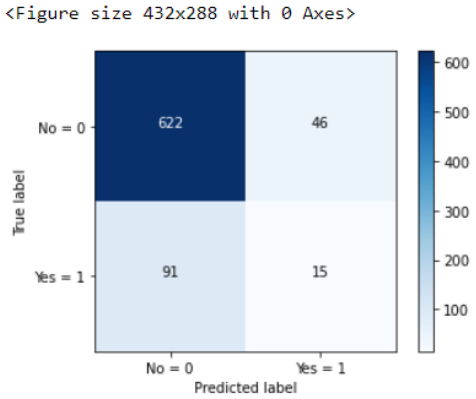
```
<Figure size 432x288 with 0 Axes>
```



(b) LR-Zero Technique

```
Train set Accuracy:  0.8948475289169295
Test set Accuracy:  0.8229974160206718
Jaccard index 0's:  0.8194993412384717
Jaccard index 1's:  0.09868421052631579
```

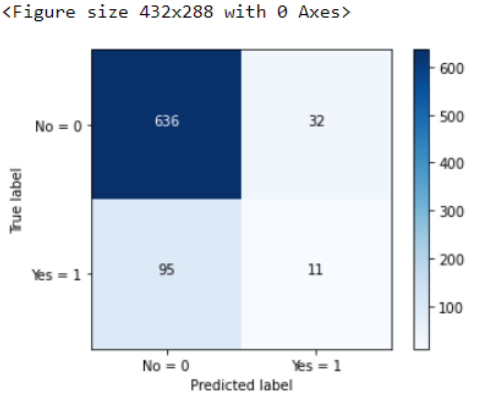
	precision	recall	f1-score	support
0.0	0.87	0.93	0.90	668
1.0	0.25	0.14	0.18	106
accuracy			0.82	774
macro avg	0.56	0.54	0.54	774
weighted avg	0.79	0.82	0.80	774



(a) LR-Mean Technique

```
Train set Accuracy:  0.8948475289169295
Test set Accuracy:  0.8359173126614987
Jaccard index 0's:  0.8335517693315858
Jaccard index 1's:  0.07971014492753623
```

	precision	recall	f1-score	support
0.0	0.87	0.95	0.91	668
1.0	0.26	0.10	0.15	106
accuracy			0.84	774
macro avg	0.56	0.53	0.53	774
weighted avg	0.79	0.84	0.80	774



(b) LR-Median Technique

Figura 5.12: Matriz de confusión de Logistic Regression

Support Vector Machines

La indagación continua con el algoritmo ***Support Vector Machine***, el cual clasifica los datos buscando un separador. Su objetivo es encontrar el hiper-plano que mejor diferencia las categorías que se quieren clasificar, pudiendo asignar cada objeto a una mitad del espacio.

Al trabajar en un espacio de alta dimensión, donde los datos se caracterizan por contar con una gran cantidad de variables, no es necesario que el conjunto de datos sea linealmente separable ya que se pueden proyectar en el espacio y encontrar un hiper-plano que los separe linealmente. Es decir, gracias a las *funciones kernel* que utiliza este algoritmo es capaz de clasificar conjuntos de datos que son linealmente separables, pero también aquellos que no lo son.

El objetivo de Support Vector Machine es encontrar el hiper-plano que mejor separe las clases en un espacio de características de alta dimensión, maximizando todo lo posible los márgenes. El hiper-plano funciona como una frontera y se busca maximizar la distancia entre los datos de cada clase y dicha frontera.

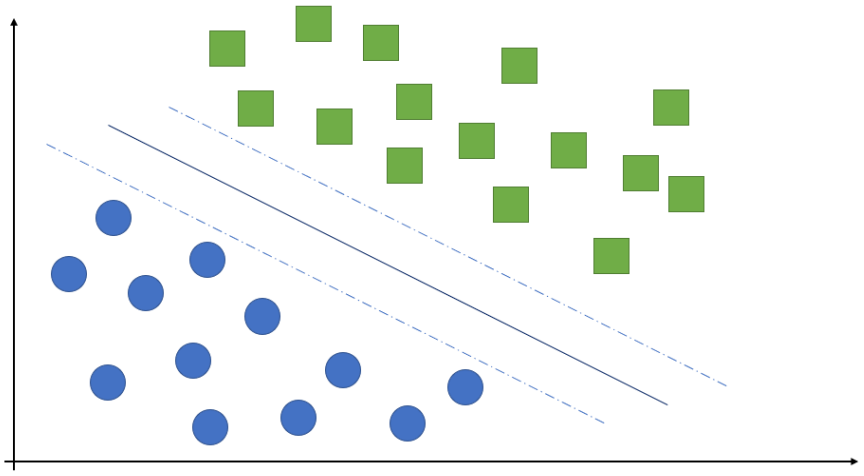


Figura 5.13: Representación algoritmo Support Vector Machine

Las distancias a maximizar se denominas margen y hay que diferenciar entre superior e inferior, los cuales se expresan matemáticamente:

$$w_1x_1 + w_2x_2 + ... + w_nx_n + b = 1 \tag{5.11}$$

$$w_1x_1 + w_2x_2 + ... + w_nx_n + b = -1 \tag{5.12}$$

Y el hiperplano (*Support Vector*):

$$w_1x_1 + w_2x_2 + ... + w_nx_n + b = 0 \tag{5.13}$$

Encontrando la combinación lineal de los valores de las variables de los datos empleados para el entrenamiento del modelo multiplicadas por sus pesos, que cumplan con las condiciones que se acaban de exponer, se tendrá un hiper-plano que maximice la distancia entre él mismo y las distintas categorías.

Si el conjunto de datos con el que se está trabajando no se puede separar linealmente, se utiliza la técnica del *Kernel*. Consiste en proyectar los datos a un espacio de características de mayor dimensión donde puedan ser separados linealmente. Gracias a esta función no lineal, los datos de entrada se proyectan a un espacio de mayor dimensión donde pueden ser separados por el hiper-plano óptimo encontrado por SVM.

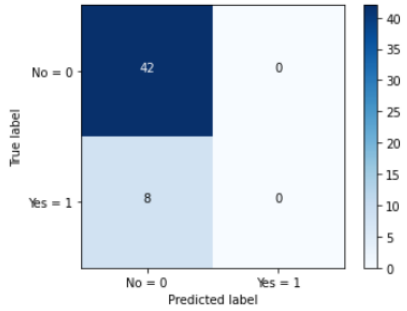
Existen diversas funciones kernel, las cuales se emplean en función de la distribución de los datos que se quieran separar. Pueden ser *Lineal, Polinómica, RBF o Sigmoidal*, entre otras. Para el desarrollo del modelo se ha escogido *RBF* cuyas siglas significan *Radial Basis Function* o *Función de Base Radial*. Esta técnica es óptima cuando la distribución de los datos que se están empleando para el entrenamiento del modelo no sigue una distribución lineal.

A continuación, se va a mostrar el resultado obtenido tras entrenar y evaluar el modelo empleando el algoritmo Support Vector Machine a cada técnica de gestión de valores nulos:

```
Train set Accuracy:  0.8769716088328076
Test set Accuracy:  0.84
Jaccard index 0's:  0.84
Jaccard index 1's:  0.0
```

	precision	recall	f1-score	support
0.0	0.84	1.00	0.91	42
1.0	0.00	0.00	0.00	8
accuracy			0.84	50
macro avg	0.42	0.50	0.46	50
weighted avg	0.71	0.84	0.77	50

```
<Figure size 432x288 with 0 Axes>
```

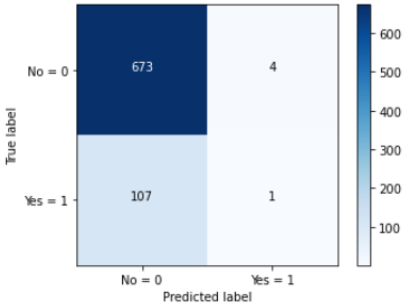


(a) SVM-Drop Technique

```
Train set Accuracy:  0.8769716088328076
Test set Accuracy:  0.858598726114649
Jaccard index 0's:  0.8584183673469388
Jaccard index 1's:  0.008928571428571428
```

	precision	recall	f1-score	support
0.0	0.86	0.99	0.92	677
1.0	0.20	0.01	0.02	108
accuracy			0.86	785
macro avg	0.53	0.50	0.47	785
weighted avg	0.77	0.86	0.80	785

```
<Figure size 432x288 with 0 Axes>
```

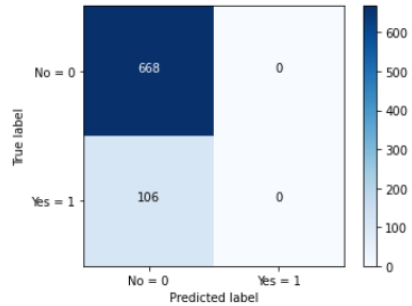


(b) SVM-Zero Technique

```
Train set Accuracy:  0.8769716088328076
Test set Accuracy:  0.8630490956072352
Jaccard index 0's:  0.8630490956072352
Jaccard index 1's:  0.0
```

	precision	recall	f1-score	support
0.0	0.86	1.00	0.93	668
1.0	0.00	0.00	0.00	106
accuracy			0.86	774
macro avg	0.43	0.50	0.46	774
weighted avg	0.74	0.86	0.80	774

```
<Figure size 432x288 with 0 Axes>
```

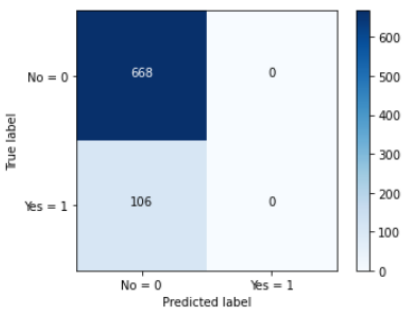


(c) SVM-Mean Technique

```
Train set Accuracy:  0.8769716088328076
Test set Accuracy:  0.8630490956072352
Jaccard index 0's:  0.8630490956072352
Jaccard index 1's:  0.0
```

	precision	recall	f1-score	support
0.0	0.86	1.00	0.93	668
1.0	0.00	0.00	0.00	106
accuracy			0.86	774
macro avg	0.43	0.50	0.46	774
weighted avg	0.74	0.86	0.80	774

```
<Figure size 432x288 with 0 Axes>
```



(d) SVM-Median Technique

Figura 5.14: Matriz de confusión de Support Vector Machines

Decision Tree & Random Forest

El siguiente algoritmo que se va a estudiar, va a basarse en la clasificación de elementos a partir de árboles de decisión. El objetivo de los ***Árboles de Decisión*** es crear un modelo que prediga el valor de una variable objetivo en función de varias variables predictoras. Es decir, el conjunto original de datos se va a ir dividiendo mientras se recorre el árbol en función de los valores de la variables predictoras. Se comprobará si cumplen con la condición impuesta en cada nodo hasta recorrer todo el árbol, terminando en un nodo hoja.

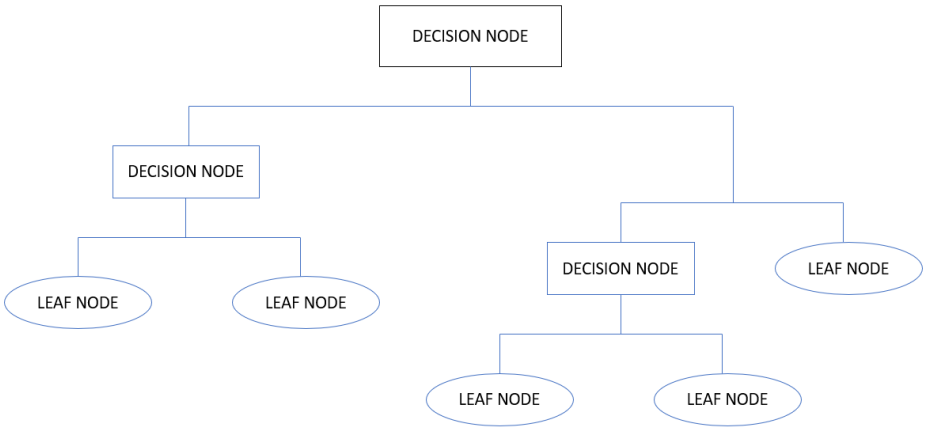


Figura 5.15: Representación algoritmo Decision Tree

El algoritmo ***Random Forest*** realiza este proceso numerosas veces, construyendo distintos árboles de decisión. Se divide el conjunto de datos dedicado al entrenamiento de forma aleatoria y cada subconjunto se emplea para entrenar un árbol de decisión distinto, formando así un "bosque".

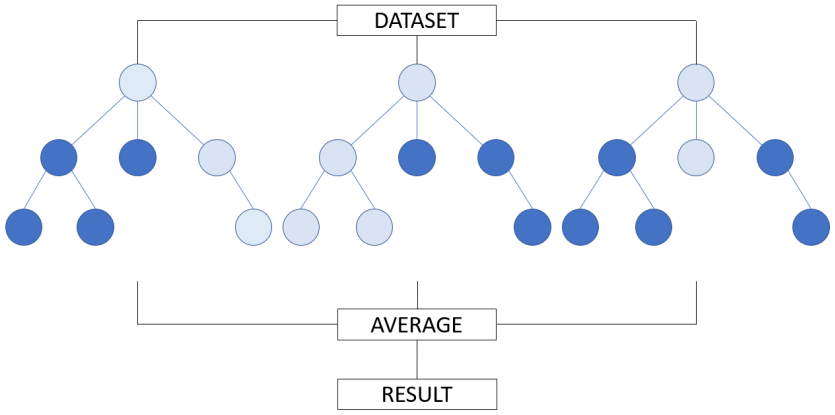


Figura 5.16: Representación algoritmo Random Forest

Si se quiere clasificar un objeto nuevo, se debe pasar a través de cada árbol entrenado y el resultado final se obtiene por mayoría, es decir, la categoría que más cantidad de veces se haya repetido será a la que corresponda el objeto a predecir.

En el modelo se han especificado tres parámetros distintos:

- **n estimators**: El número de árboles de decisión que se van a construir.
- **max depth**: El número máximo de variables predictoras que va a tener cada árbol.
- **min samples leaf**: El número mínimo de muestras necesarias para ser un nodo hoja.

La elección de de los tres parámetros se ha realizado comparando de forma exhaustiva verificando con que valor se obtenía un mejor resultado.

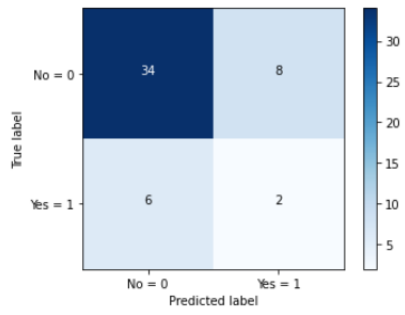
Se van a mostrar los modelos descritos a partir de ambos algoritmos, con el objetivo de comparar los resultados obtenidos. Cabe destacar que el criterio de impureza escogido ha sido la entropía, la cual calcula el nivel de entre los datos que pertenecen a una misma categoría.

A continuación, se va a mostrar el resultado obtenido tras entrenar y evaluar el modelo empleando el algoritmo Árboles de Decisión a cada técnica de gestión de valores nulos. Seguido, se va a mostrar el resultado obtenido empleando el algoritmo Random Forest:

```
Train set Accuracy:  0.8696109358569927
Test set Accuracy:  0.72
Jaccard index 0's:  0.7083333333333334
Jaccard index 1's:  0.125
```

	precision	recall	f1-score	support
0.0	0.85	0.81	0.83	42
1.0	0.20	0.25	0.22	8
accuracy			0.72	50
macro avg	0.53	0.53	0.53	50
weighted avg	0.75	0.72	0.73	50

```
<Figure size 432x288 with 0 Axes>
```

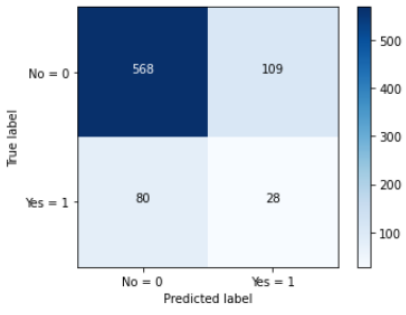


(a) DT-Drop Technique

```
Train set Accuracy:  0.8696109358569927
Test set Accuracy:  0.759235668789809
Jaccard index 0's:  0.750330250990753
Jaccard index 1's:  0.12903225806451613
```

	precision	recall	f1-score	support
0.0	0.88	0.84	0.86	677
1.0	0.20	0.26	0.23	108
accuracy			0.76	785
macro avg	0.54	0.55	0.54	785
weighted avg	0.78	0.76	0.77	785

```
<Figure size 432x288 with 0 Axes>
```



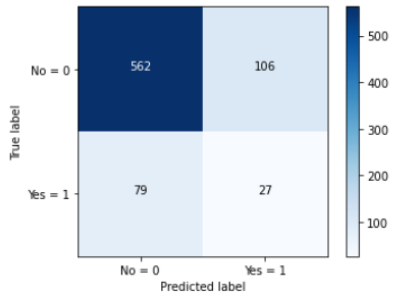
(b) DT-Zero Technique

Figura 5.17: Matriz de confusión de Decision Tree

```
Train set Accuracy:  0.8696109358569927
Test set Accuracy:  0.7609819121447028
Jaccard index 0's:  0.7523427041499331
Jaccard index 1's:  0.12735849056603774
```

	precision	recall	f1-score	support
0.0	0.88	0.84	0.86	668
1.0	0.20	0.25	0.23	106
accuracy			0.76	774
macro avg	0.54	0.55	0.54	774
weighted avg	0.78	0.76	0.77	774

```
<Figure size 432x288 with 0 Axes>
```

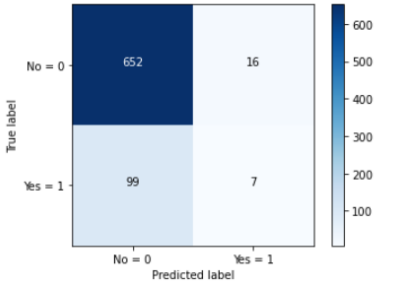


(a) DT-Mean Technique

```
Train set Accuracy:  0.8696109358569927
Test set Accuracy:  0.851421188630491
Jaccard index 0's:  0.8500651890482399
Jaccard index 1's:  0.05737704918032787
```

	precision	recall	f1-score	support
0.0	0.87	0.98	0.92	668
1.0	0.30	0.07	0.11	106
accuracy			0.85	774
macro avg	0.59	0.52	0.51	774
weighted avg	0.79	0.85	0.81	774

```
<Figure size 432x288 with 0 Axes>
```

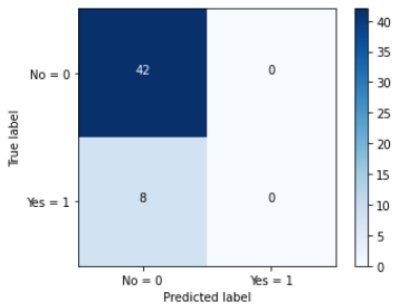


(b) DT-Median Technique

```
Train set Accuracy:  0.8727655099894848
Test set Accuracy:  0.84
Jaccard index 0's:  0.84
Jaccard index 1's:  0.0
```

	precision	recall	f1-score	support
0.0	0.84	1.00	0.91	42
1.0	0.00	0.00	0.00	8
accuracy			0.84	50
macro avg	0.42	0.50	0.46	50
weighted avg	0.71	0.84	0.77	50

```
<Figure size 432x288 with 0 Axes>
```

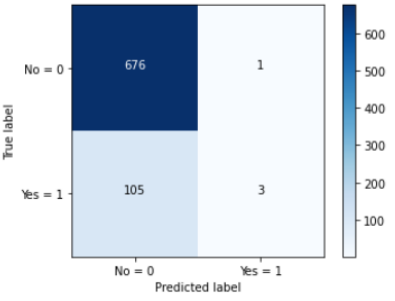


(a) RF-Drop Technique

```
Train set Accuracy:  0.8785488958990536
Test set Accuracy:  0.8649681528662421
Jaccard index 0's:  0.8644501278772379
Jaccard index 1's:  0.027522935779816515
```

	precision	recall	f1-score	support
0.0	0.87	1.00	0.93	677
1.0	0.75	0.03	0.05	108
accuracy			0.86	785
macro avg	0.81	0.51	0.49	785
weighted avg	0.85	0.86	0.81	785

```
<Figure size 432x288 with 0 Axes>
```

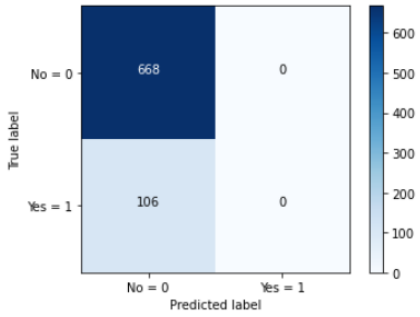


(b) RF-Zero Technique

```
Train set Accuracy:  0.8785488958990536
Test set Accuracy:  0.8630490956072352
Jaccard index 0's:  0.8630490956072352
Jaccard index 1's:  0.0
```

	precision	recall	f1-score	support
0.0	0.86	1.00	0.93	668
1.0	0.00	0.00	0.00	106
accuracy			0.86	774
macro avg	0.43	0.50	0.46	774
weighted avg	0.74	0.86	0.80	774

<Figure size 432x288 with 0 Axes>

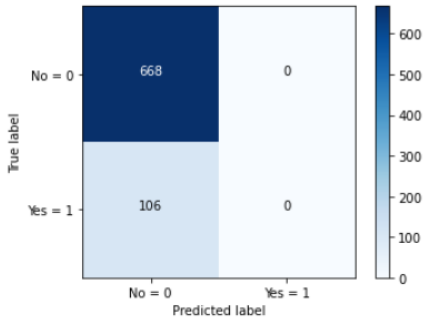


(a) RF-Mean Technique

```
Train set Accuracy:  0.8769716088328076
Test set Accuracy:  0.8630490956072352
Jaccard index 0's:  0.8630490956072352
Jaccard index 1's:  0.0
```

	precision	recall	f1-score	support
0.0	0.86	1.00	0.93	668
1.0	0.00	0.00	0.00	106
accuracy			0.86	774
macro avg	0.43	0.50	0.46	774
weighted avg	0.74	0.86	0.80	774

<Figure size 432x288 with 0 Axes>



(b) RF-Median Technique

Figura 5.20: Matriz de confusión de Random Forest

5.1.6. Análisis de los resultados

Con el objetivo de valorar como de bien se comportan los modelos expuestos y elegir cual se va a aplicar para la detección de dislexia se han empleado una serie de técnicas para evaluar el rendimiento de los modelo, determinando el grado de precisión y exactitud con la que predicen los valores.

- **Matriz de confusión**: Se utiliza para evaluar el rendimiento de un modelo de clasificación. Se trata de una matriz diferenciada entre los valores reales y los valores predichos por el modelo. De esta manera es capaz de plasmar la cantidad de valores clasificados correctamente y los que se clasificaron incorrectamente en cada clase.

- **Train Accuracy vs Test Accuracy**: La precisión (*accuracy*) se emplea para evaluar el rendimiento de un modelo de clasificación, al igual que la Matriz de confusión. Se va a diferenciar entre *Train Accuracy* y *Test Accuracy*, donde el primer término hace referencia a la proporción de valores de entrenamiento que se clasificaron correctamente y el segundo a la proporción de valores de prueba que se clasificaron correctamente. Se pretende obtener una precisión en el conjunto de prueba lo más alta posible, ya que esto implicaría que el modelo es capaz de predecir adecuadamente los valores introducidos.

- **Jacard Index**: Se utiliza para evaluar la similitud entre dos conjuntos de datos. Se encarga de

evaluar como de similares son los conjuntos de datos clasificados como positivos con respecto al conjunto de datos que verdaderamente son positivos.

- **Precision**: Indica la proporción de valores clasificados como positivos y que realmente son positivos. La precisión es útil cuando se quieren minimizar los falsos positivos.
- **Recall**: Indica la proporción de valores positivos que se han clasificado correctamente por el modelo de clasificación diseñado. El recall es útil cuando se pretenden minimizar los falsos negativos.
- **F1 score**: Es una medida que combina la precisión y el recall. El F1-score se calcula como la media armónica de la precisión y el recall. Se trata de una medida útil cuando se quiere encontrar un equilibrio entre la precisión y el recall, es decir, se emplea para evaluar si el modelo es capaz de identificar correctamente la mayoría de los valores pertenecientes a una categoría sin clasificar incorrectamente demasiados valores que no pertenecen a dicha categoría.

Teniendo en consideración el conjunto de medidas de evaluación de los modelos de clasificación y los resultados obtenidos se pueden obtener diferentes conclusiones.

En primer lugar, es importante destacar que la técnica de manejo de valores nulos que elimina objetos que contienen características con valores no numéricos no es recomendable, dado que reduce significativamente las dimensiones del *dataset*. Como resultado el conjunto de datos resultante es mucho más pequeño y no representativo de la población. Aunque dicha técnica solo se ha aplicado al *dataset* de testeo, al reducirlo cuatro veces su tamaño original, las dimensiones de los conjuntos de datos no cumplen con las reglas empleadas para el diseño de un modelo en Machien Learning.

Cabe destacar, que tras eliminar los *outliers* las dimensiones se ven reducidas considerablemente, resultando en un conjunto de datos para el testeo que cuenta con 50 datos frente a 1902 para el entrenamiento. Es por esto por lo que los modelos en los que se ha empleado la eliminación de valores nulos no se van a tener en cuenta como modelos válidos.

Valorando la comparación entre *Train Accuracy* y *Test Accuracy* se puede observar como los resultados obtenidos son muy positivos, puesto que todos superan el 72 % de rendimiento. No obstante, estos resultados no son del todo precisos, ya que aunque sí que cuentan con niveles de predicción elevados, esto se debe a la facilidad de predecir los valores negativos. Es decir, el modelo ha sido entrenado satisfactoriamente para detectar los casos negativos en dislexia, sin embargo, los casos positivos apenas son reconocidos correctamente.

La razón detrás de este fenómeno se debe a la presencia de una gran cantidad de casos negativos en dislexia en comparación con los casos positivos. Al ser la mayoría de valores negativos, el modelo se entrena para categorizar correctamente la mayor cantidad de instancias posibles, obteniendo como resultado modelos especializados en clasificar correctamente aquellos objetos que posean la etiqueta *Dyslexia = 'No'*

Las medidas de precisión presentan valores muy elevados debido al calcular la media entre todos los datos y, teniendo en cuanta el fenómeno mencionado, la gran mayoría resultan satisfactoriamente clasificados. Sin embargo, esta resolución no es la deseada, ya que se busca un modelo capaz de predecir de similar manera ambas categorías.

Si se analizan las medidas *Jacard Index, Precision, Recall y F1 score* diferenciando entre la clase 0.0 y 1.0 se corrobora las conclusiones obtenidas en esta sección. Mientras que para la primera clase (0.0) los resultados son excelentes, obteniendo medidas que alcanzan el 100 %, la segunda clase (1.0) no se identifica rigurosamente, ya que los resultados completamente opuestos. Por ello se necesitan implementar medidas capaces de gestionar la desproporción de la clase minoritaria.

5.1.7. Gestión de la desproporción entre clases

En la etapa de *Entrenamiento y Testeo del modelo* se resaltó el hecho de que el número de personas que no tienen dislexia (0.0) es mucho mayor en comparación con los que sí tienen dislexia (1.0). Aunque la muestra refleja la población real, este desequilibrio ha derivado en modelos que no son capaces de predecir los casos positivos. Para combatir la desproporción entre las dos clases en las que se pueden clasificar los datos se van a emplear distintas técnicas, las cuales pretenden romper con el desequilibrio entre los valores del conjunto de datos:

- **Undersampling**: Consiste en eliminar aleatoriamente algunas instancias de la clase mayoritaria hasta lograr un equilibrio con la clase minoritaria [59].
- **Oversampling**: Consiste en replicar aleatoriamente algunos ejemplos de la clase minoritaria hasta lograr un equilibrio con la clase mayoritaria [59].
- **SMOTE**: Consiste en la interpolación de los casos existentes con el objetivo de crear nuevas instancias de la clase minoritaria. Se seleccionan aleatoriamente instancias pertenecientes a la clase minoritaria, se buscan sus vecinos más cercanos en el espacio de características y se crean nuevos ejemplos en el espacio entre el ejemplo original y sus vecinos [60].

Estas tres técnicas se han implementado a las tres combinaciones de los *dataset* de forma paralela, resultando en:

$$\text{Modelos} = 5 \text{ Algoritmos} \times 4 \text{ Técnicas Nulos} \times 3 \text{ Comb. Dataset} \times 3 \text{ Técnicas Desequilibrio} \quad (5.14)$$

El objetivo es encontrar el modelo que mejor se adapte a los requisitos deseados, prediciendo en similar medida los casos negativos y positivos. A continuación, se va a exponer la implementación de cada técnica.

Undersampling

El Submuestreo (*undersampling*) es una técnica de balanceo de datos. Consiste en reducir el número de instancias de la clase mayoritaria (*Dyslexia = 'No'*) de forma aleatoria a las dimensiones de la clase minoritaria (*Dyslexia = 'Yes'*). De esta forma se tendrá el mismo número de instancias positivas como negativas y se alcanzará un equilibrio entre ambas.

La gestión de desproporción de las clases se ha realizado en el paso *Limpieza y preparación de datos*. Una vez convertidos todos los valores a numéricos, se ha modificado la dimensión de los *dataset*:

```
#class count
count_dyslexia_no, count_dyslexia_yes = data.Dyslexia.value_counts()

#divide by class
dyslexia_no_df = data[data['Dyslexia'] == 0]
```

```
dyslexia_yes_df = data[data['Dyslexia'] == 1]
#undersample 0-class
dyslexia_no_df = dyslexia_no_df.sample(count_dyslexia_yes)
#concat the df both class
data = pd.concat([dyslexia_no_df, dyslexia_yes_df], axis=0)
```

Listing 5.6: Gestión de desproporción de las clases mediante Submuestreo

Se puede observar como con la librería *RandomOverSampler* las dimensiones del *sub-dataset* que contiene los valores negativos se ha reducido al número de datos del *sub-dataset* compuesto por valores positivos.

El procedimiento realizado tras la conversión de las dimensiones del *dataset* es el mismo que el descrito anteriormente.

Oversampling

El Sobremuestreo (*oversampling*) es otra técnica de balanceo de datos. Consiste en aumentar el número de instancias de la clase minoritaria (*Dyslexia = 'Yes'*) duplicándolas de forma aleatoria con el objetivo de alcanzar las dimensiones de la clase mayoritaria (*Dyslexia = 'No'*). Se conseguirá tener el mismo número de instancias positivas como negativas y se alcanzará un equilibrio entre ambas.

Al igual que en *Undersampling* la gestión de desproporción de las clases se ha realizado en el paso *Limpieza y preparación de datos*. Una vez convertidos todos los valores a numéricos, se ha modificado la dimensión de los *dataset*:

```
#class count
count_D_no, count_D_yes = data.Dyslexia.value_counts()

#divide by class
D_no_df = data[data['Dyslexia'] == 0]
D_yes_df = data[data['Dyslexia'] == 1]

#oversample 1-class
D_yes_df = D_yes_df.sample(count_D_no, replace = True)
#concat df both class
data = pd.concat([D_no_df, D_yes_df], axis=0)
```

Listing 5.7: Gestión de desproporción de las clases mediante Submuestreo

Es necesario comprobar que los índices de las instancias que forman los conjuntos de datos sean únicos. Al haber replicado valores ya existentes su índices se han podido copiar, por ello es conveniente re-ajustarlos:

```
data = data.reset_index(drop=True)
print("Duplicated data from Data Set: ",data.index.duplicated().sum())
```

Mediante la librería RandomOverSampler se han aumentado el número de instancias del *sub-dataset* que contiene los valores positivos hasta igualar el número de instancias del *sub-dataset* que contiene los valores negativos.

SMOTE

SMOTE se trata de una técnica de sobremuestreo, al igual que *oversampling*, empleada para afrontar el problema de desequilibrio de clases. Se basa en la generación de nuevas instancias, a partir de la interpolación, de la clase minoritaria para igualar el número de instancias de la clase mayoritaria.

La nueva instancia se genera calculando la diferencia entre la instancia seleccionada y uno de los K vecinos seleccionados, multiplicándola por un número aleatorio entre 0 y 1 y sumándola al vecino elegido. Esto permite generar nuevos objetos a partir de valores ya existentes, lo que ayuda a evitar la producción de instancias completamente aleatorias que podrían introducir ruido en el conjunto de datos. A diferencia de las dos técnicas de balanceo de clases, no se implementa en el paso *Limpieza y preparación de datos*, si no que se aplica al separar entre el *dataset* con todas las variables y el que contiene los valores relativos a la dislexia:

```
from sklearn.model_selection import train_test_split
from imblearn.over_sampling import SMOTE

y = data['Dyslexia']
X = data.loc[:, data.columns != 'Dyslexia']

smote = SMOTE(sampling_strategy = 'minority')
X_sm, y_sm = smote.fit_resample(X, y)
```

Listing 5.8: gestión de desproporción de las clases mediante SMOTE

De esta forma se va a conseguir que la clase minoritaria no sea ignorada por el modelo debido a sus bajas dimensiones. La generación de instancias nuevas ayuda a aumentar el tamaño de la clase minoritaria y, por lo tanto, permitirá mejorar el rendimiento del modelo en la predicción de dicha clase.

5.1.8. Elección del modelo final

Finalmente, se va a elegir el modelo que mejores resultados proporcione. Se va a analizar cada caso derivando en la opción que mejor cumpla con los requisitos que se desea que tenga el modelo de Machine Learning.

Comparando los valores obtenidos tras calcular el *Accuracy* con el conjunto dedicado al entrenamiento y el dedicado al testeo se obtiene, en función de la combinación de los conjuntos de datos empleados:

Desktop and Tablet Dataset

ACCURACY		K-Nearest Neighbors				Logistic Regression				Support Vector Machines				Decision Trees				Random Fosrest			
		Drop	Null	Mean	Median	Drop	Null	Mean	Median	Drop	Null	Mean	Median	Drop	Null	Mean	Median	Drop	Null	Mean	Median
Train set	Imbalanced data	0.8980	0.8980	0.8980	0.8980	0.9175	0.9175	0.9048	0.9048	0.8707	0.8707	0.8707	0.8707	0.8833	0.8833	0.8833	0.8833	0.9001	0.9001	0.9001	0.9043
	Undersampling	0.8402	0.8475	0.8475	0.8475	0.8475	0.8475	0.8475	0.8475	0.8378	0.8378	0.8378	0.8378	0.8281	0.8281	0.8281	0.8281	0.9274	0.9346	0.9322	0.9322
	Oversampling	0.9572	0.9572	0.9567	0.9997	0.8536	0.8536	0.8536	0.8536	0.9048	0.9048	0.9048	0.9048	0.7850	0.7850	0.7850	0.7850	0.8536	0.8536	0.8536	0.8536
	SMOTE	0.8657	1.0	0.8385	0.9967	0.8633	0.8615	0.8615	0.8612	0.9079	0.9083	0.9134	0.9125	0.8596	0.8584	0.8627	0.8550	0.9669	0.9667	0.9636	0.9679
Test set	Imbalanced data	0.84	0.8459	0.8527	0.8553	0.8	0.8344	0.8269	0.8307	0.84	0.7261	0.8630	0.8630	0.84	0.6446	0.8617	0.8617	0.84	0.6427	0.8630	0.8630
	Undersampling	0.5	0.4651	0.5726	0.5726	0.4615	0.5969	0.5385	0.6324	0.4808	0.6047	0.4274	0.4103	0.4808	0.6512	0.6410	0.6410	0.5577	0.6512	0.4615	0.5214
	Oversampling	0.4747	0.4752	0.4321	0.3920	0.7532	0.6512	0.5871	0.5781	0.4367	0.5903	0.5051	0.5051	0.7152	0.5182	0.62	0.62	0.7532	0.6512	0.5871	0.5781
	SMOTE	0.6667	0.5945	0.5748	0.5501	0.7262	0.6211	0.5397	0.5838	0.5	0.5480	0.5007	0.5104	0.5	0.5591	0.5763	0.5816	0.8333	0.6928	0.5898	0.5299

Cuadro 5.7: Comparación de la medida *Accuracy* empleando el conjunto de datos Desktop y Tablet

Tablet Dataset

ACCURACY		K-Nearest Neighbors				Logistic Regression				Support Vector Machines				Decision Trees				Random Fosrest			
		Drop	Null	Mean	Median	Drop	Null	Mean	Median	Drop	Null	Mean	Median	Drop	Null	Mean	Median	Drop	Null	Mean	Median
Train set	Imbalanced data	1.0	0.8901	0.8821	0.8821	0.9	0.8837	0.8934	0.8917	0.825	0.8631	0.8626	0.8626	1.0	0.8981	0.9030	0.8998	0.825	0.8758	0.8788	0.8853
	Undersampling	0.7561	0.7739	0.7699	0.7787	0.9024	0.7826	0.8318	0.8318	0.5609	0.7130	0.6902	0.7079	1.0	0.8869	0.8141	0.8141	0.9268	0.9130	0.8673	0.8761
	Oversampling	0.9965	1.0	0.9977	0.9977	0.9056	0.7843	0.7820	0.7835	0.8776	0.7835	0.7543	0.7505	0.9405	0.7559	0.7677	0.7812	1.0	0.9403	0.9520	0.9423
	SMOTE	1.0	0.9991	1.0	1.0	0.9552	0.8014	0.8117	0.8014	0.7462	0.8070	0.7256	0.7340	1.0	0.7987	0.8380	0.8286	0.9402	0.9630	0.9700	0.9747
Test set	Imbalanced data	0.9	0.8598	0.8645	0.8645	0.8	0.8599	0.8452	0.8581	0.9	0.8599	0.8645	0.8645	0.5	0.8662	0.8516	0.8516	0.9	0.8599	0.8645	0.8645
	Undersampling	0.9091	0.5862	0.4827	0.4827	0.8181	0.6552	0.4827	0.5172	0.1818	0.5172	0.5862	0.5517	0.7272	0.5862	0.5172	0.5862	0.8181	0.5862	0.5517	0.4827
	Oversampling	0.9583	0.9552	0.9371	0.9401	0.8333	0.7821	0.7605	0.7635	0.75	0.7403	0.7245	0.7506	0.8333	0.7403	0.7604	0.7754	0.9305	0.9074	0.9520	0.9042
	SMOTE	0.9411	0.8265	0.8544	0.8358	1.0	0.7380	0.8059	0.7574	1.0	0.7158	0.7052	0.7014	0.8823	0.7527	0.8171	0.7761	0.9411	0.8819	0.9402	0.9029

Cuadro 5.8: Comparación de la medida *Accuracy* empleando el conjunto de datos Tablet

Desktop Dataset

ACCURACY		K-Nearest Neighbors	Logistic Regression	Support Vector Machines	Decision Trees	Random Fosrest
Train set	Imbalanced data	0.9057	0.9050	1.0	0.8895	0.8936
	Undersampling	0.8788	0.8509	1.0	0.8757	0.8322
	Oversampling	1.0	0.8790	1.0	0.9441	0.9970
	SMOTE	0.9996	0.8685	1.0	0.9105	0.9894
Test set	Imbalanced data	0.8548	0.8844	0.8521	0.8763	0.8736
	Undersampling	0.6666	0.7777	0.7530	0.7654	0.6543
	Oversampling	0.9550	0.8731	0.9564	0.9273	0.9550
	SMOTE	0.8385	0.8416	0.9388	0.8699	0.8871

Cuadro 5.9: Comparación de la medida *Accuracy* empleando el conjunto de datos Desktop

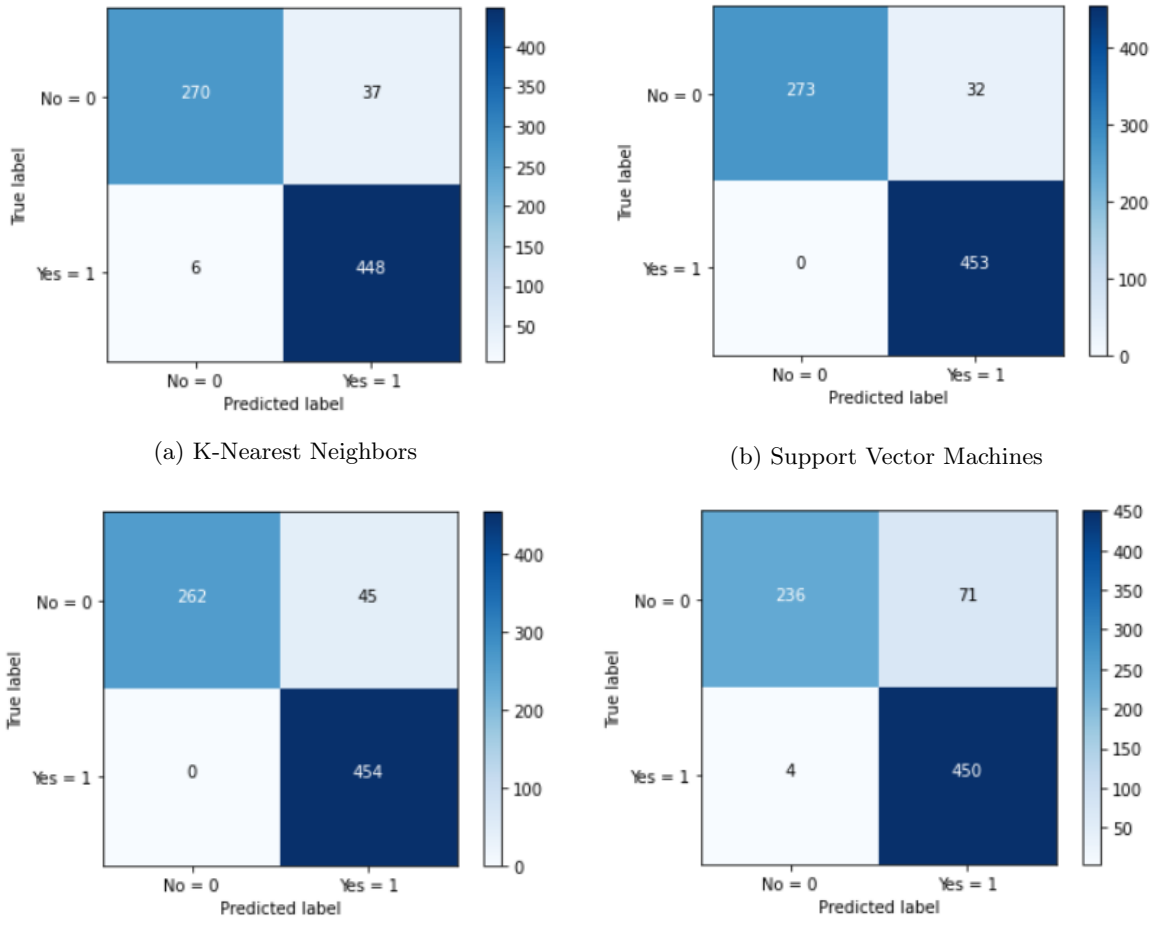
Se puede observar como es necesaria la gestión de la desproporción de clases, debido a la falta de precisión y exactitud al predecir la clase positiva (*Dyslexia = 'Yes'*). Para ello se han empleado técnicas de submuestreo y sobremuestreo, donde se disminuyen o aumentan las dimensiones del *dataset* con la intención de equilibrar las dos clases. Tras aplicar las diferentes técnicas los resultados de los modelos mejoraron, sin embargo, no todos los modelos eran válidos.

Tras implementar ***Undersampling*** las dimensiones de los *dataset* disminuyeron notablemnte, ya que las instancias negativas se redujeron al número de instancias positivas. Al no contar con apenas objetos, los modelos no eran capaces de predecir correctamente ambas clases. Aunque los valores predichos y los valores reales aumentaron en similitud, no era suficiente como para afirmar que los modelos diseñados eran buenos. Por ello los resultados obtenidos tras implementar el submuestreo se descartaron.

A continuación, se implementaron las técnicas de sobremuestreo ***Oversampling*** y ***SMOTE***. Con ellas se obtuvieron resultados más positivos. No obstante, es necesario diferenciar entre los distintos

subconjuntos empleados para el diseño del modelo. Empleando los *dataset* Desktop para el entrenamiento y Tablet para el testeo el modelo contaba con demasiados datos. Esto desembocó en un *Train Accuracy* muy elevado, ya que el modelo se adaptaba adecuadamente al conjunto de datos de entrenamiento, sin embargo, el *Test Accuracy* era demasiado bajo ya que los datos de testeo no se modelaban correctamente. Este fenómeno recibe el nombre de *sobreajuste* y por ello estos modelos se descartaron.

Por lo tanto el modelo se diseñó empleando el *dataset Desktop*, el cual se dividió dedicando parte al entrenamiento (80 %) y parte al testeo (20 %). Aunque se podría haber elegido el *dataset Tablet*, se consideró mejor elección el primero al no contar con valores nulos y al poseer mayores dimensiones. Entre las técnicas de *sobreajuste y SMOTE*, la que mejores resultados dio fue la primera, eligiendo el algoritmo de *Support Vector Machine*:



(a) K-Nearest Neighbors

(b) Support Vector Machines

(c) Decision Tree

(d) Random Forest

Figura 5.21: Matriz de confusión del modelo final

5.2. Desarrollo de la aplicación web

La aplicación web se ha construido empleando el framework Django, el cual permite diseñar aplicaciones web de manera rápida y eficiente utilizando Python. Tal y como se mencionó en la sección anterior el modelo de Machine Learning se ha desarrollado en Python, por ello era necesario trabajar con un framework que permitiese integrar la interfaz de usuario, para la cual se ha empleado HTML, CSS y JavaScript, y el modelo mencionado de la forma más simple posible. A su vez, Django permite administrar el sitio web e incluye un sistema de autenticación.

A su vez, Django destaca por su sistema de plantillas, el cual permite crear interfaces de usuario de manera eficiente y escalable. Las plantillas son archivos que contienen HTML y otros elementos de diseño (CSS, JS, imágenes...) que permiten definir la estructura y el contenido de una página web. Incluyen etiquetas y filtros los cuales ayudan a reutilizar trozos de código en diferentes partes de la aplicación, añadiendo lógica y contenido dinámico. Además, Django incluye soporte para herencia, lo que permite a los desarrolladores definir una plantilla base y construir el resto de vistas sobre la misma.

En este proyecto, estas ventajas se han aprovechado al máximo con el objetivo de desarrollar una aplicación web eficiente y escalable. La página web está estructurada principalmente en:

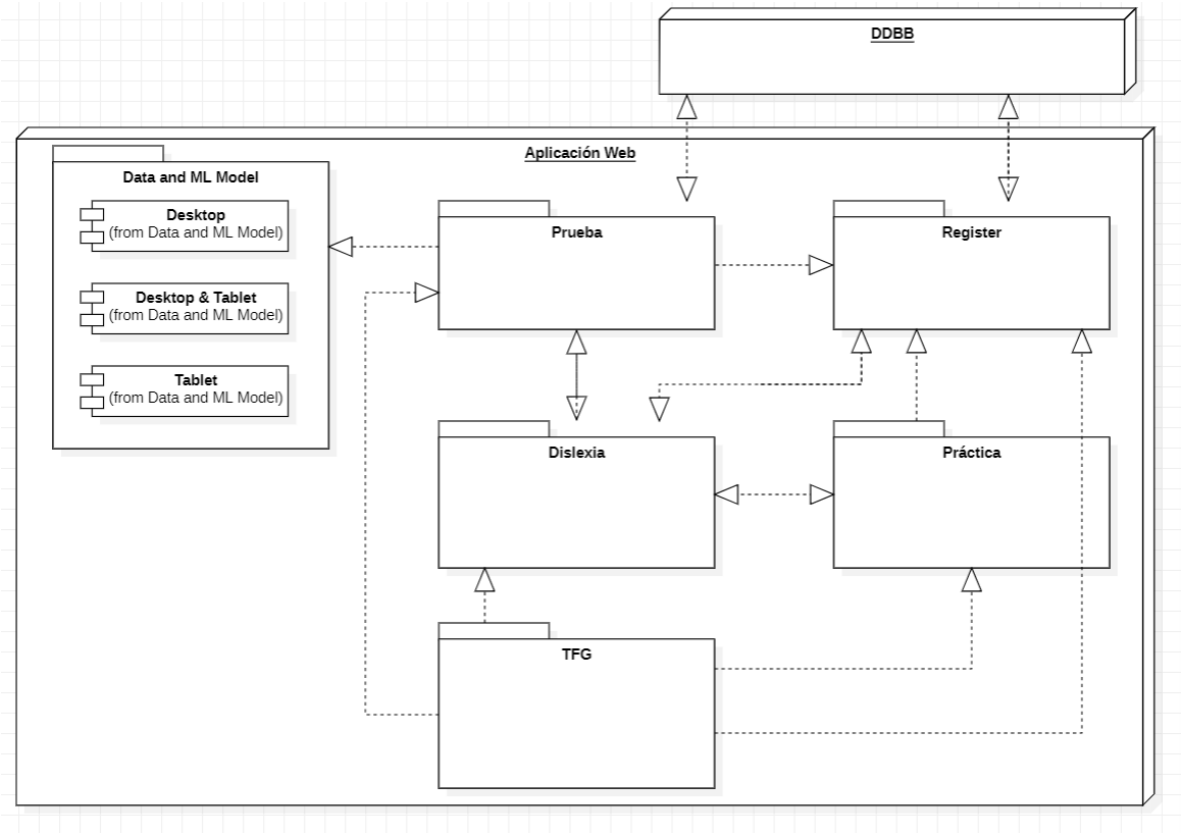


Figura 5.22: Diagrama de componentes del proyecto

Como se muestra en la Figura 6.1, el proyecto se ha estructurado mediante múltiples directorios, ya que se ha querido sacar el mayor partido a las aplicaciones que ofrece Django. Las aplicaciones o *applications*[6] que ofrece Django son módulos autocontenidos, es decir, unidades de código que representan una funcionalidad específica del proyecto web.
De esta forma se ha conseguido separar la lógica asociada a cada módulo del proyecto, consiguiendo que, por ejemplo, la gestión de usuarios y la prueba de detección de dislexia sean independientes entre sí, al igual que ocurre con el resto de aplicaciones.

Ha resultado especialmente útil puesto que han permitido dividir el proyecto en secciones más pequeñas y manejables, facilitando el desarrollo, el testeo y la reutilización del código. A continuación, se va a introducir brevemente cada división realizada:

- **Data & ML Model**: Este directorio no corresponde a una aplicación de Django, sino que se encarga de alojar los distintos modelos de Machine Learning explicados en la sección anterior. Es importante destacar que está compuesto por tres carpetas, cada una asociada a una combinación distinta de los conjuntos de datos (*Datasets*). Además, contiene un fichero llamado *ml_model.py*, el cual incluye el código de Machine Learning que se empleará en la aplicación web para predecir la dislexia.

- **Dyslexia**: La aplicación *Dyslexia* contiene las vistas principales de la aplicación web. Corresponde al punto de partida para acceder a las distintas funcionalidades que se ofrecen, como el test de detección o la práctica adicional, así como el proceso completo de registro y autenticación e información que puede resultar relevante para el usuario.

- **Práctica**: La aplicación *Práctica* contiene las vistas correspondientes a los ejemplos de ejercicios que se proponen para tratar la dislexia en un ambiente dinámico. Está compuesta por seis actividades que se focalizan en distintos procesos cognitivos. Para acceder a los ejercicios es necesario que el usuario esté autenticado.

- **Prueba**: La aplicación *Prueba* contiene las vistas correspondientes al test de detección de dislexia. Al acceder al test se mostrará una vista con la información necesaria para realizarlo y, seguidamente, se accederá a las 32 cuestiones que lo conforman, junto con sus enunciados correspondientes. Finalmente, se mostrará el resultado obtenido, indicando si el usuario es o no disléxico, así como las puntuaciones y los procesos cognitivos trabajados en cada ejercicio. Para el acceso a la prueba de detección de dislexia es necesario que el usuario esté atenticado.

- **Register**: La aplicación *Register* contiene las vistas correspondientes al manejo y autenticación del usuario. Los usuarios podrán crear una cuenta y, en caso de que ya tengan una, iniciar sesión para acceder al contenido de la aplicación. A su vez, podrán acceder a la información de su perfil y modificarla, así como eliminar la cuenta, cambiar la contraseña o recuperarla mediante el correo asociado.

- **Static**: Este directorio no corresponde a una aplicación de Django, sino que almacena archivos estáticos de la página web, como JavaScript, CSS, imágenes y otros recursos que no se modifican dinámicamente durante la ejecución de la aplicación.

- **TFG**: Este directorio es el de configuración del proyecto. Contiene archivos esenciales para el funcionamiento de la aplicación web, destacando el archivo *settings.py*, que aloja la configuración global del proyecto, como la de la base de datos o la seguridad. Por otro lado, también cuenta con *urls.py*, que define las rutas del proyecto.

[6] https://docs.djangoproject.com/en/4.2/ref/applications/

Una vez que se ha analizado la estructura del proyecto, se profundizará en los procesos llevados a cabo durante el desarrollo de la aplicación web. A continuación, se muestra el diagrama de casos de uso de la aplicación web en la Figura 5.23:

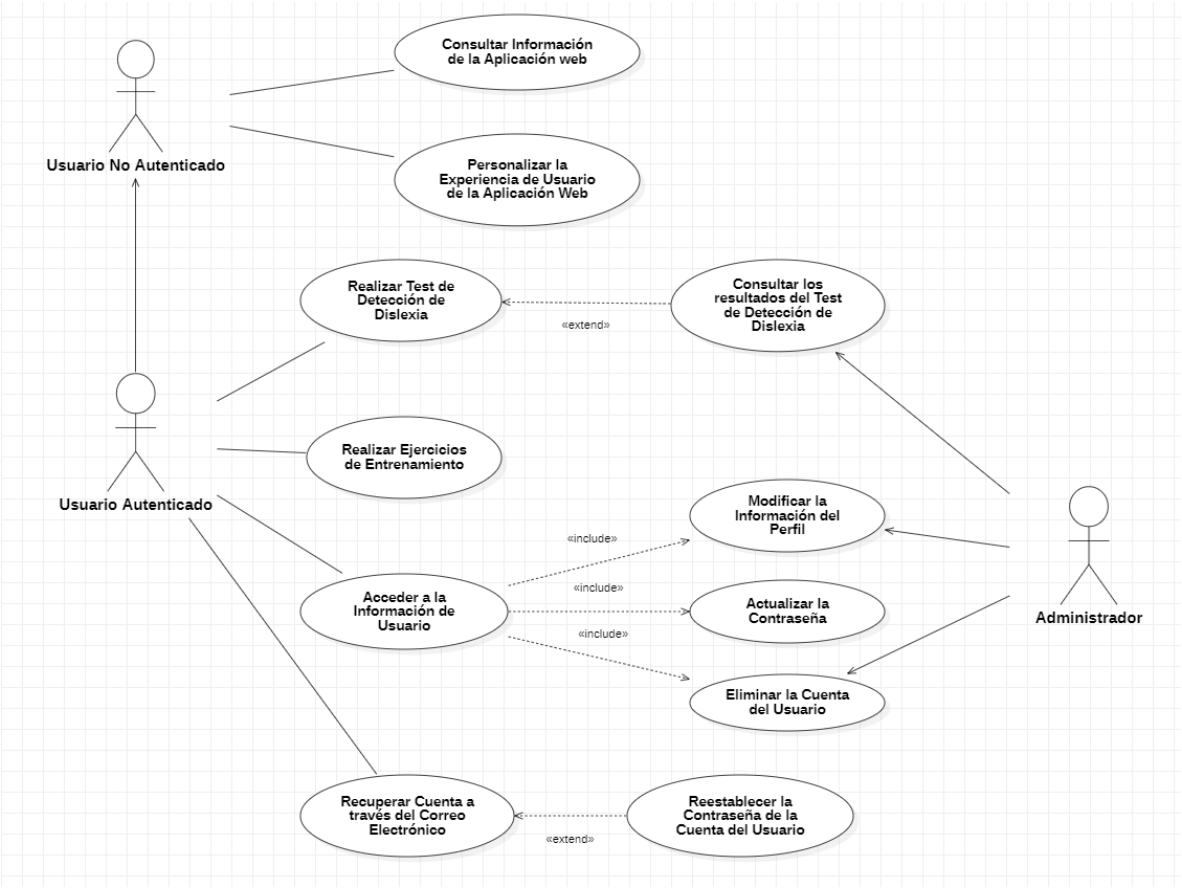


Figura 5.23: Diagrama de Casos de Uso de la Aplicación Web

Se distinguen tres actores: el usuario no autenticado, el usuario autenticado y el administrador. Se puede observar como el administrador tiene acceso a la información personal del usuario. Tiene permisos para realizar modificaciones en la información de la cuenta, siendo capaz incluso de eliminarla. A su vez, se puede observar como las funcionalidades de la aplicación se reservan para los usuarios autenticados, aunque los no autenticados pueden acceder a las secciones informativas de la aplicación, así como modificar los ajustes de accesibilidad.

5.2.1. Autenticación

La autenticación en la aplicación web ha recibido gran importancia, puesto que mantener la seguridad y privacidad de los datos de los usuarios es esencial en cualquier sitio web. Se han utilizando las funcionalidades proporcionadas por Django.

Django incluye un sistema de gestión de usuarios que permite crear, modificar y eliminar nuevos usuarios. A partir del objeto "User" proporcionado por Django, el cual incluye campos como nombre de usuario, contraseña o dirección de correo electrónico, se ha creado un modelo adicional llamado "Profile". Este modelo recopila toda la información personal necesaria para alimentar el modelo de Machine Learning y obtener resultados válidos, como la edad o el sexo del usuario.

```
from django.db import models
from django.contrib.auth.models import AbstractUser

class Profile(AbstractUser):
    first_name = models.CharField(max_length=50)
    last_name = models.CharField(max_length=50)
    email = models.EmailField(max_length=150,default="")
    edad = models.PositiveSmallIntegerField(null=True)
    sexo = models.CharField(max_length=10, choices=[("Hombre"), ("Mujer")])
    lengua = models.CharField(max_length=50, choices=[("Si"),("No")])
    suspender = models.CharField(max_length=50, choices=[("Si"),("No")])
```

Listing 5.9: Clase *Profile* en models.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from .models import Profile

class ProfileForm(UserCreationForm):

    class Meta:
        model = Profile
        fields = ["first_name", "last_name", "username", "email", "edad", "
            sexo", "lengua suspender', "password1", "password2"]

        labels = {
            'first_name': 'Nombre',
            'last_name': 'Apellidos',
            'lengua': ' Hablas  solo un idioma o eres biling$ $e?',
            'suspender':' Has  suspendido alguna asignatura relacionada con el
                habla castellano?'}
```

Listing 5.10: Clase *ProfileForm* en forms.py

A continuación, se va a mostrar de forma gráfica **el registro y el inicio de sesión** que deben realizar los usuarios para acceder a todas las funcionalidades que ofrece la web. Se va a mostrar de forma visual como se conectan los tres bloques principales que conforman toda aplicación web: *la interfaz de usuario*, por la cual el usuario introduce su información personal, *el procesamiento de la aplicación*, que recibe los datos del usuario y los maneja de forma segura, y *la base de datos*, que gestiona y almacena los datos de la aplicación web.

En ambos casos, tanto en el inicio de sesión como en el registro de una nueva cuenta, el usuario debe introducir la información que se especifica por pantalla y enviarla. El sistema la recibirá, realizará una petición *POST* y si se comprueba que el formulario es válido, obtendrá un mensaje de éxito (*200*)

indicando que la acción se ha realizado correctamente. El usuario se habrá autenticado y podrá acceder a todas las funcionalidades que se le ofrecen.

Registro de un Usuario Nuevo

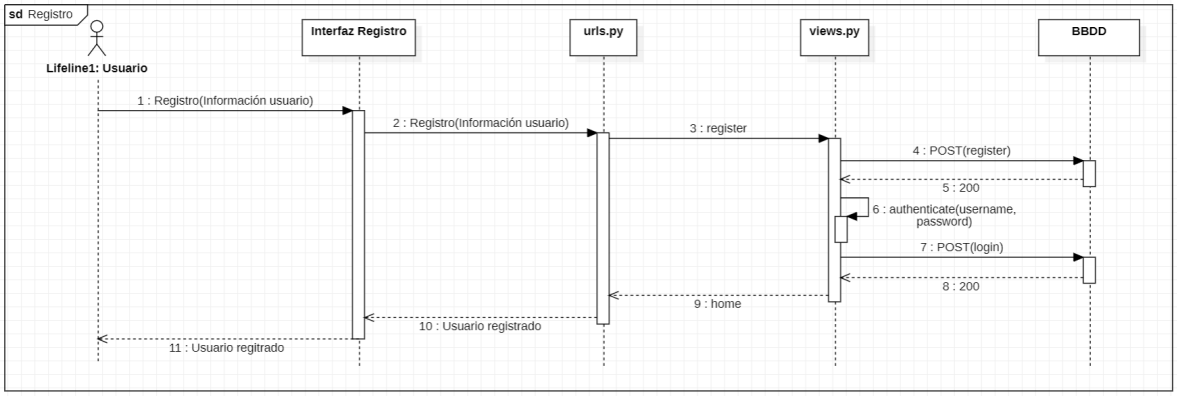


Figura 5.24: Diagrama de Secuencias Registro de un Usuario Nuevo

Inicio de Sesión

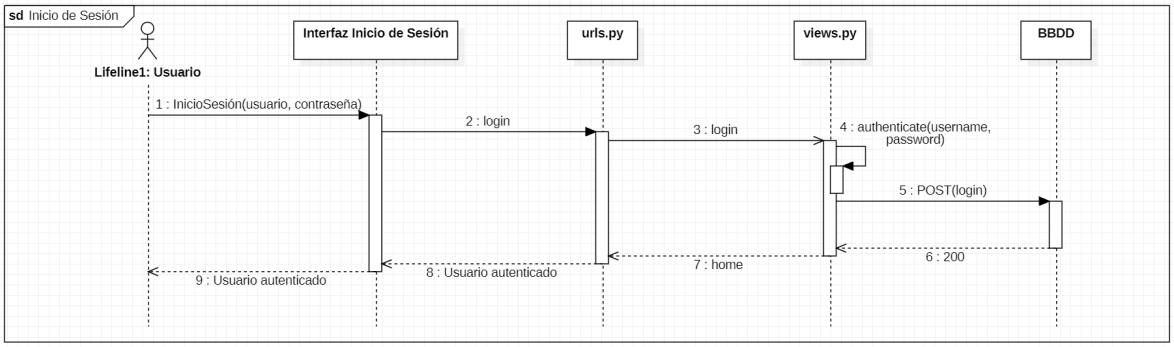


Figura 5.25: Diagrama de Secuencias Inicio Sesión

Las contraseñas deben cumplir una serie de características para considerarse válidas. Mediante *AUTH PASSWORD VALIDATORS* [7] Django permite definir una lista de validadores de contraseñas con el objetivo de garantizar que las contraseñas cumplan con ciertos criterios de seguridad y complejidad. La lista incluye validadores de similitud con contraseñas demasiado comunes, como *Password* o *1234*, o con la información personal que ha proporcionado el usuario. A su vez, verifican que la longitud sea de ocho caracteres mínimo y que no sea enteramente numérica.

[7] https://docs.djangoproject.com/en/4.1/ref/settings/auth-password-validators

Figura 5.26: Registro de usuario

En caso de que no cumpla alguno de los cuatro requisitos se proporcionará un mensaje de error indicando claramente cual es el problema y el usuario deberá introducir una nueva contraseña:

Figura 5.27: Mensaje de error en el Registro del Usuario

Cada vez que los usuarios inicien sesión o se registren por primera vez podrán acceder a su perfil y modificar la información que se les presenta. Cada campo se muestra de forma individual, claramente diferenciado por una etiqueta, junto a un botón (*icono lapicero*). Si se desea modificar la información que se presenta en el perfil, basta con reescribirla y pulsar dicho botón para que se actualice en la base de datos. A continuación, se muestra el diagrama de secuencias correspondiente a la actualización de la información del perfil del usuario:

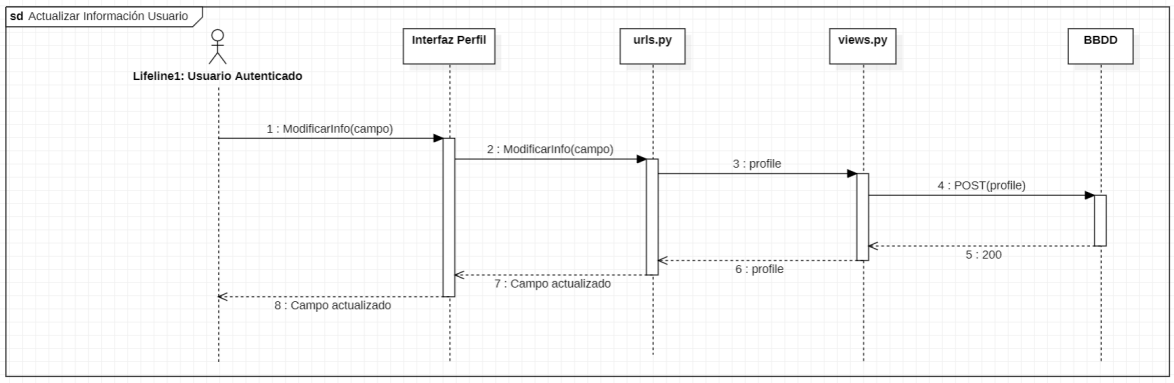


Figura 5.28: Diagrama de Secuencias Actualización de la Información del Perfil de Usuario

La interfaz que se muestra al usuario, descrita anteriormente, es:

Figura 5.29: Perfil del Usuario

A su vez, se ofrecen dos opciones adicionales en la vista correspondiente al perfil de usuario: "Eliminar cuenta" y "Restablecer contraseña". Como sugieren sus nombres, la primera opción permite eliminar las cuentas asociadas a los usuarios. Al seleccionar esta opción, aparecerá un modal en pantalla que requerirá la confirmación del usuario. Si se confirma, la información personal del usuario se eliminará de la base de datos.

Eliminar cuenta

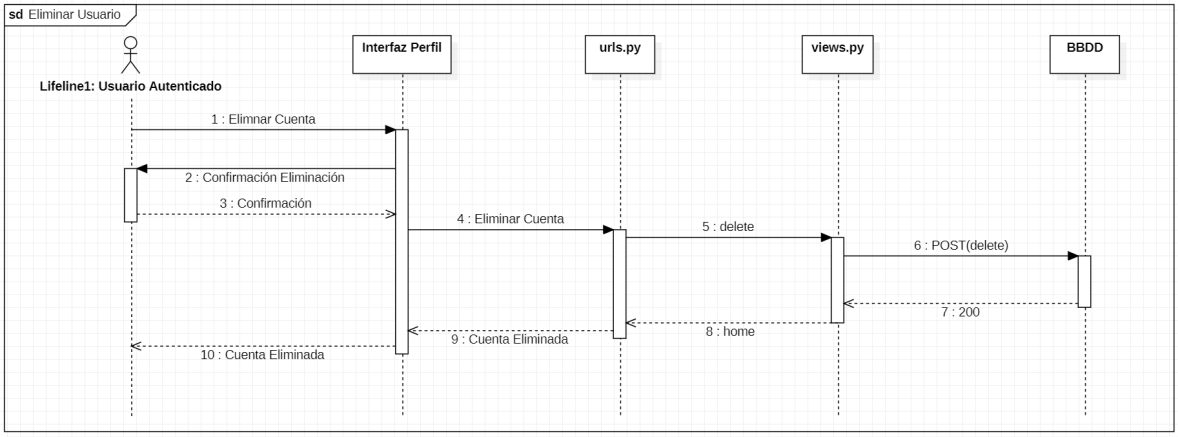


Figura 5.30: Diagrama de Secuencias Eliminar Cuenta del Usuario

Por otra parte, la segunda opción permite restablecer la contraseña que el usuario emplea para acceder a la página web. De esta forma, se facilita la actualización periódica de la contraseña, manteniendo la

cuenta segura. El usuario deberá introducir la contraseña original y, a continuación, la nueva contraseña, la cual deberá cumplir con los mismos requisitos que se exigían durante el registro.

Restablecer contraseña

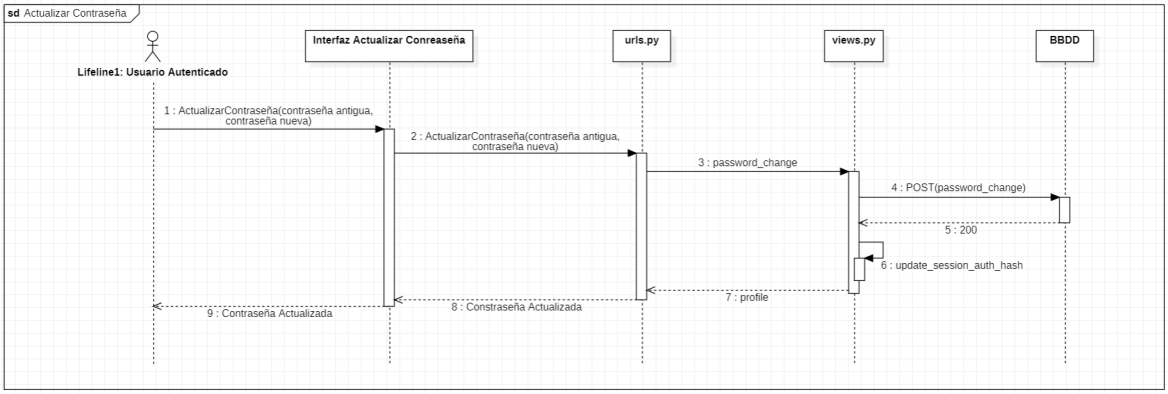


Figura 5.31: Diagrama de Secuencias Actualizar Contraseña

La interfaz que se le muestra al usuario para cambiar la contraseña es:

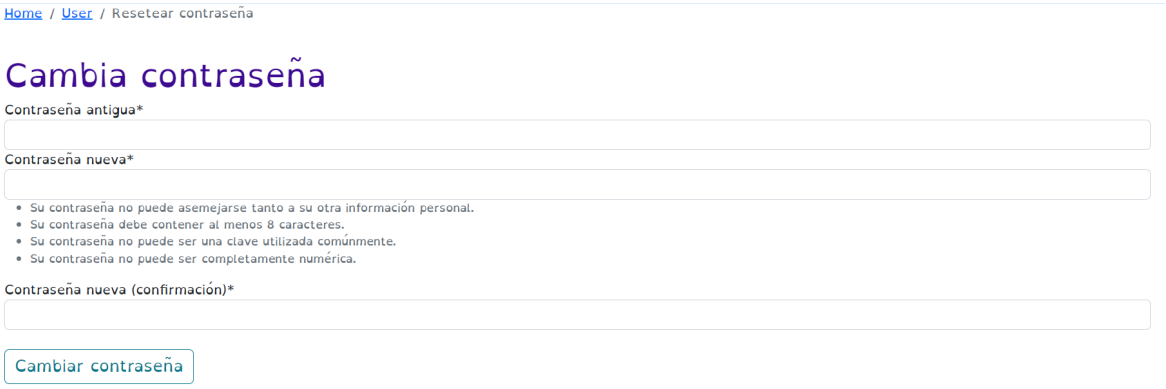


Figura 5.32: Restablecer la contraseña

En caso de que un usuario haya olvidado su contraseña y no pueda acceder a su cuenta, se ofrece la opción de recuperarla. La implementación de esta funcionalidad mejora la experiencia de usuario al proporcionar una solución rápida y efectiva a un problema común. Además, ayuda a mantener la confianza de los usuarios en la plataforma, ya que no se ven forzados a crear una nueva cuenta para continuar utilizando el servicio en caso de que no recuerden la contraseña. Por último, al permitir a los usuarios recuperar sus contraseñas de manera segura, se garantiza la protección de su información personal y se reduce el riesgo de acceso no autorizado a sus cuentas.

En la vista de inicio de sesión, se ofrecen dos enlaces: "Crear cuenta nueva" y "Restablecer la contraseña". Al acceder al segundo enlace, aparece en pantalla la opción de introducir el correo electrónico asociado a la cuenta. Es importante que el usuario tenga un correo electrónico preestablecido, ya que

el sistema buscará en la base de datos dicho correo y modificará la cuenta de usuario que coincida. Si no se tiene un correo asociado a la cuenta, no se podrá recuperar la contraseña de forma segura.

Figura 5.33: Paso 1 para recuperar la cuenta

Tras dar a enviar se mostrará por pantalla un mensaje informando de que se ha enviado un correo al correo proporcionado.

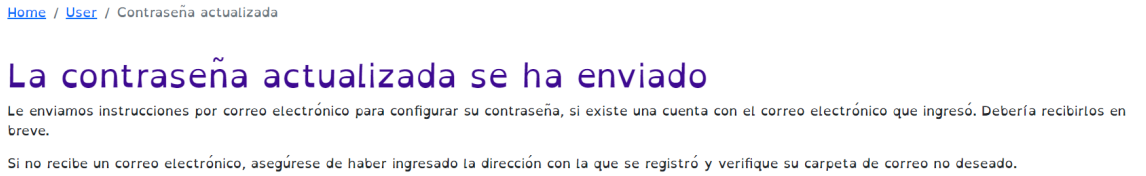


Figura 5.34: Paso 2 para recuperar la cuenta

Una vez se reciba dicho correo, será necesario seguir el enlace proporcionado.

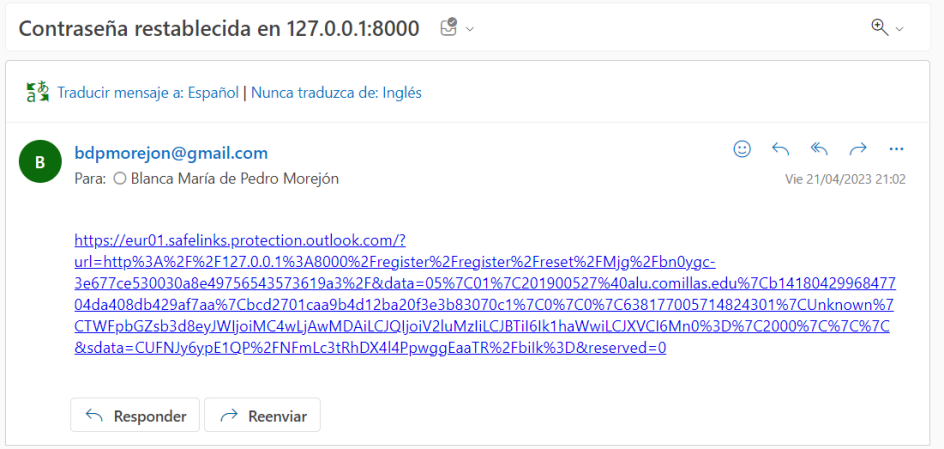


Figura 5.35: Correo para recuperar la cuenta

En enlace redirige al usuario a una nueva vista de la aplicación web que le permitirá introducir una contraseña nueva, siempre respetando las validaciones de autenticación ya mencionadas en las secciones anteriores.

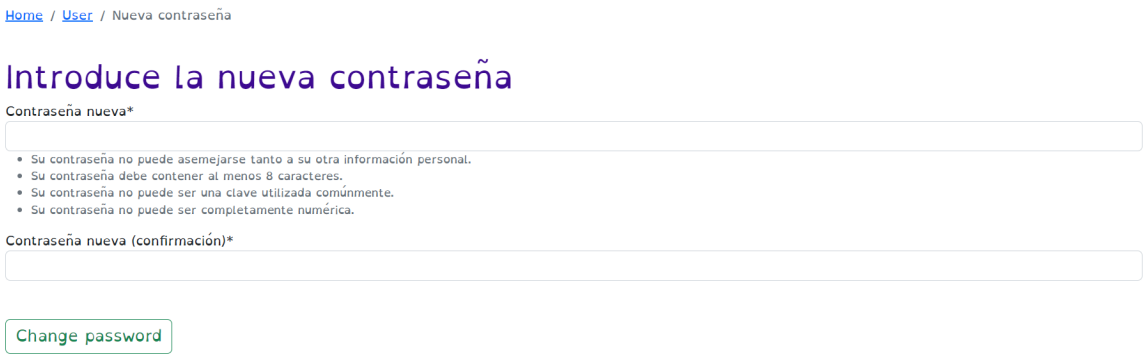


Figura 5.36: Paso 3 para recuperar la cuenta

Finalmente, se mostrará un mensaje de éxito, el cual permitirá acceder al inicio de la aplicación e iniciar sesión con la nueva configuración.

Figura 5.37: Paso 4 para recuperar la cuenta

El diagrama de secuencias que corresponde a la descripción de los pasos que se acaban de exponer:

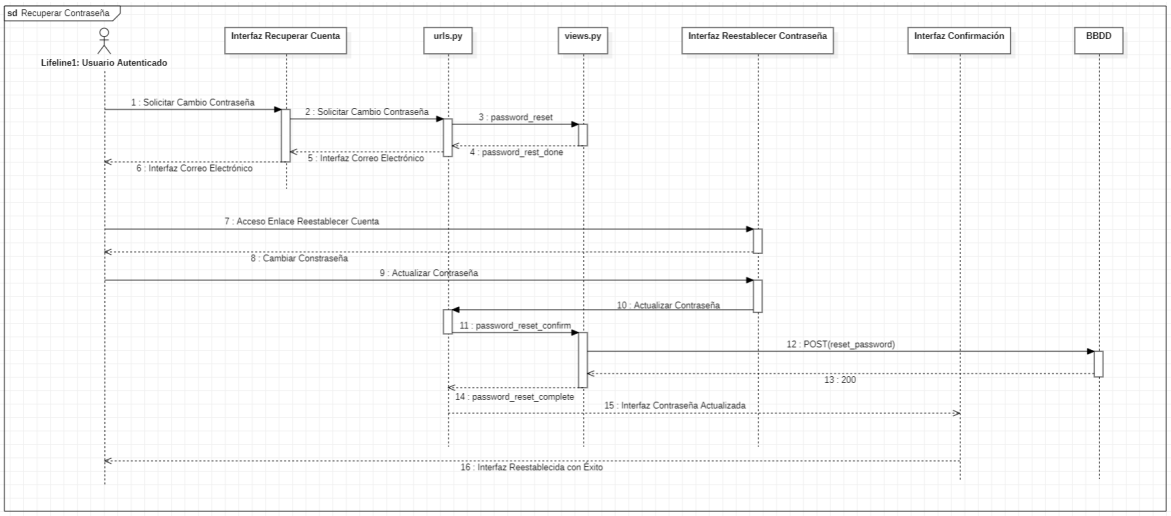


Figura 5.38: Diagrama de Secuencias Reestablecer Cuenta de Usuario

5.2.2. Test de Detección de Dislexia

Como se lleva mencionando hasta el momento, una de las principales funcionalidades que ofrece la aplicación web es el test de detección de dislexia.

Esta prueba está compuesta por 32 ejercicios, los cuales siguen todos la misma estructura. Al usuario se le presenta el enunciado en formato audio y cuando esté preparado podrá acceder al ejercicio pulsando el botón "Acceso al ejercicio". Seguido, se le mostrará correspondiente del test y tendrá un tiempo limitado para su realización. Cuando dicho tiempo se agote, se mostrará un panel por pantalla indicándole que el ejercicio ha finalizado, impidiéndole continuar, y se especificará la puntuación obtenida. El usuario deberá pulsar el botón "Continuar", para acceder al siguiente ejercicio, hasta que se complete la prueba y pueda visualizar los resultados finales.

El diagrama de secuencias correspondiente a la realización de un ejercicio genérico del test es:

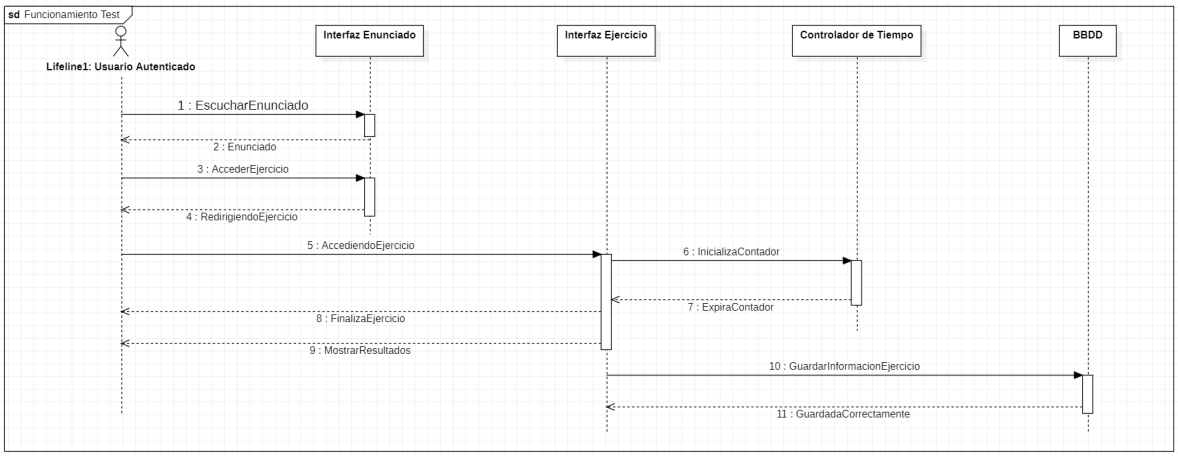


Figura 5.39: Diagrama de Secuencia de un ejercicio genérico del Test

La lógica de cada uno de los ejercicio del test se ha planteado en el Anexo D: Test de detección de dislexia, en el que se detalla cada uno de los ejercicios.

Al usuario se le muestran 32 vistas distintas para cada cuestión, diferenciando entre el enunciado y el ejercicio, junto con las correspondientes al inicio del test y a los resultados finales. Sin embargo, en Django solo se han establecido 5 rutas correspondientes a:

- **intro**: Ruta que permite acceder a la introducción del Test donde se especifican las instrucciones para su realización.
- **instructions/<int:index>**: Ruta que permite acceder a la vista base de cada enunciado de las preguntas del test, diferenciada por el índice de cada pregunta.
- **question/<int:index>**: Ruta que permite acceder a la vista base de cada ejercicio de las preguntas del test, diferenciada por el índice de cada pregunta.
- **save_test_results/<int:user_profile_id>/<int:index>/**: Ruta que permite almacenar los resultados de un usuario concreto (*user_profile_id*) de cada pregunta del test, diferenciadas por su índice.

- **results/<int:user_profile_id>**: Ruta que permite acceder a los resultados de un usuario concreto (*user_profile_id*) tras la realización del ejercicio.

urls.py

```
from django.urls import path
from . import views

app_name = "prueba"
urlpatterns = [
    path("intro", views.intro, name="intro"),
    path("instructions/<int:index>", views.instructions, name="instructions"),
    path("results/<int:user_profile_id>", views.results, name="results"),
    path("question/<int:index>", views.question, name="question"),
    path('save_test_results/<int:user_profile_id>/<int:index>/', views.
        save_test_results, name='save_test_results')]
```

Listing 5.11: Definición de las rutas de la *App práctica* en urls.py

El proceso que se le presenta al usuario se describe a continuación con un diagrama de flujo:

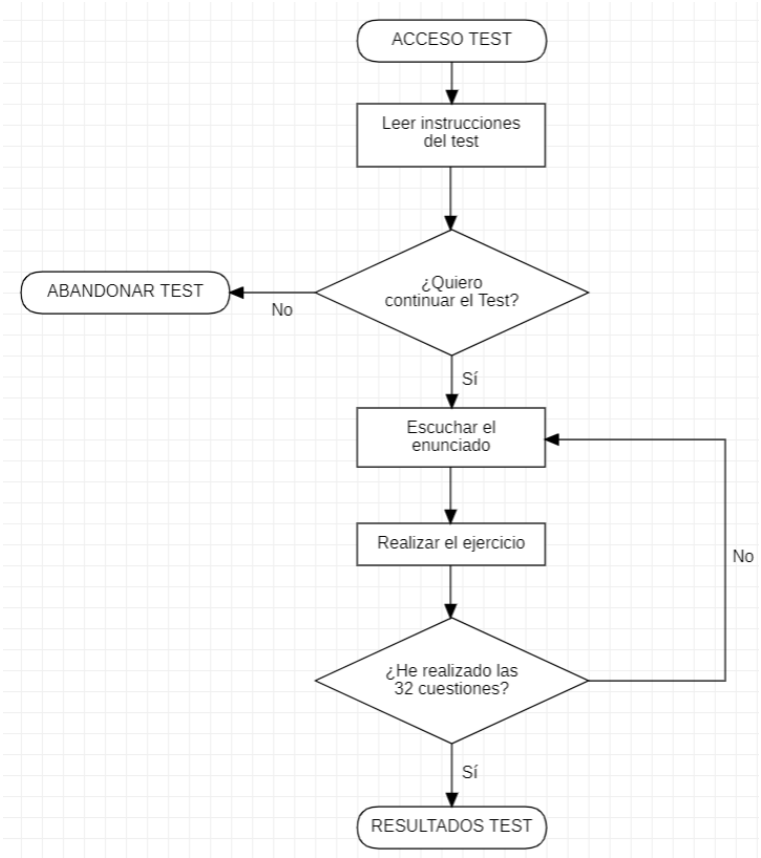


Figura 5.40: Diagrama de Flujo del Test de Detección de Dislexia

5.2.3. Ejercicios de Entrenamiento de Dislexia

A su vez, la aplicación ofrece distintos ejercicios que puede realizar un usuario autenticado para mejorar ciertas habilidades. Entre ellos se encuentran: el juego del ahorcado, el juego de la memoria, el tres en raya, el twister, un ejercicio de asociación de imágenes con la palabra correspondiente y en ejercicio de reordenación de palabras. La lógica detrás de cada ejercicio se expone en el Anexo E: Ejercicios de entrenamiento.

A continuación, se va a exponer como se presenta cada ejercicio al usuario mediante un diagrama de secuencias. El usuario deberá elegir que ejercicio quiere practicar entre la seis opciones que se le ofrecen y realizar lo que se le indica. Una vez haya finalizado, tendrá la opción de seguir practicado y reiniciar el ejercicio o abandonarlo.

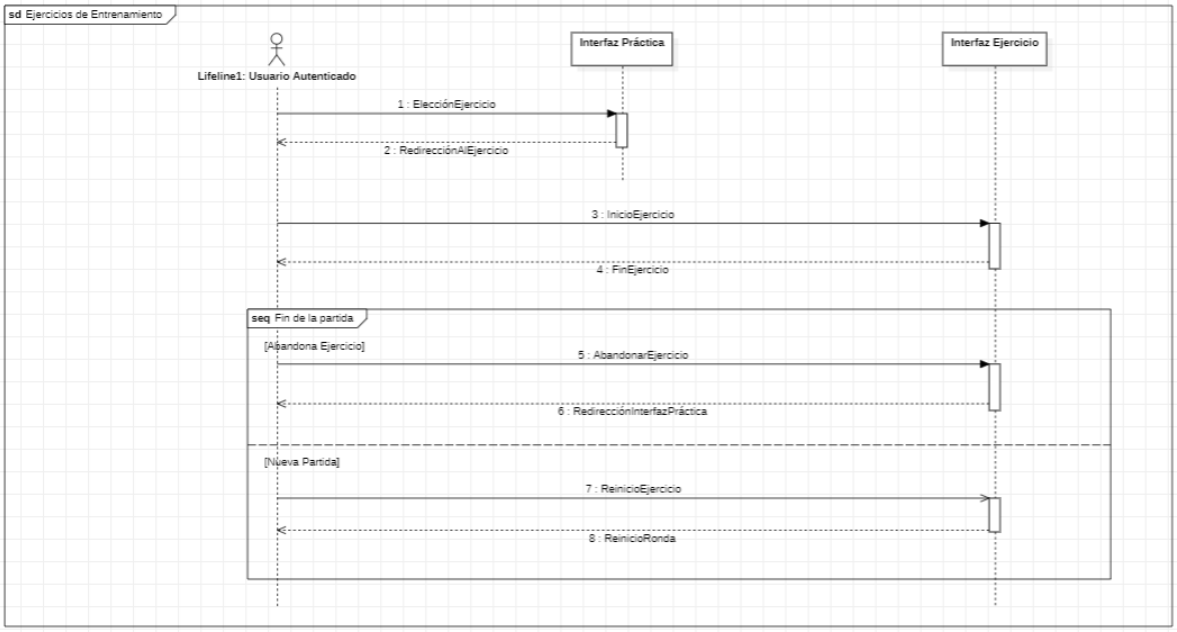


Figura 5.41: Diagrama de Secuencia de un ejercicio genérico de entrenamiento

Django ha permitido que mediante dos rutas se pueda implementar y diseñar la funcionalidad correspondiente a esta sección:

- **home**: Ruta que permite acceder a la página de inicio de la aplicación "practica" definida en Django.
- **exercise/<int:index>**: Ruta que permite acceder a la vista base de cada ejercicio de entrenamiento, diferenciada por el índice correspondiente.

urls.py

```
from django.urls import path
from . import views

app_name = "practica"
urlpatterns = [
    path("home", views.home, name="home"),
    path("exercise/<int:index>", views.exercise, name="exercise"),
]
```

Listing 5.12: Definición de las rutas de la *App prueba* en urls.py

A pesar de la simplicidad que presenta la conexión de las diferentes rutas que conforman la aplicación, los usuario podrán moverse por los ejercicios con total libertad, tal y como se muestra en el siguiente diagrama de flujo:

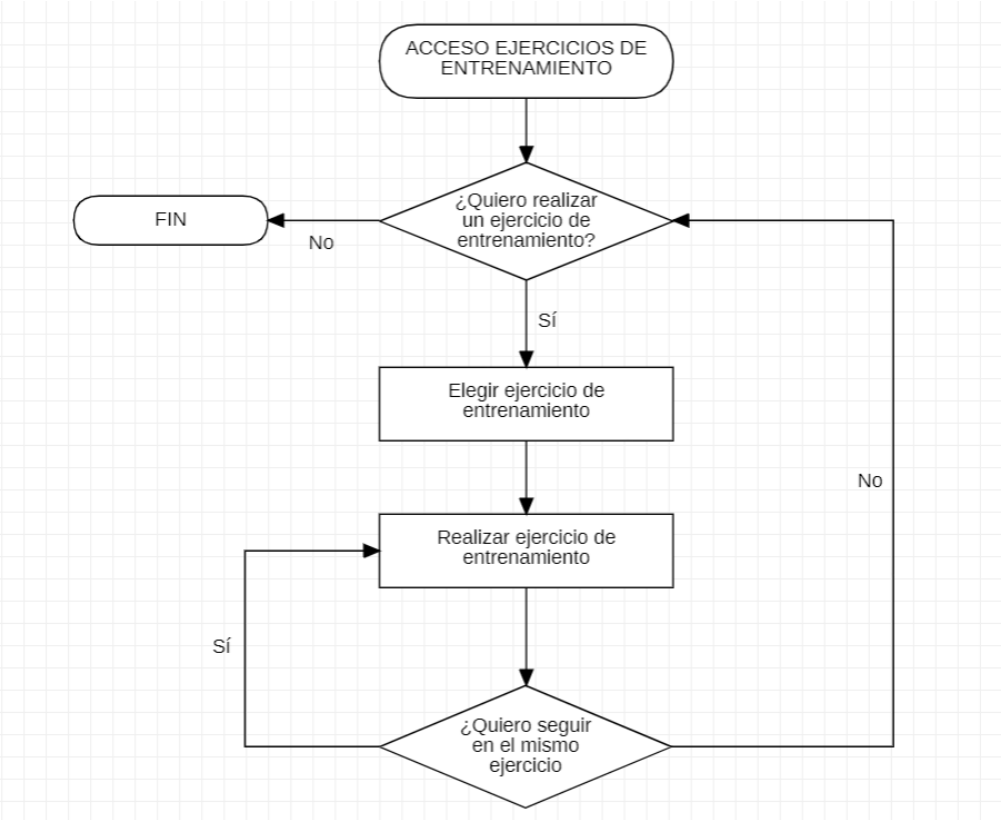


Figura 5.42: Diagrama de Flujo de los Ejercicios de Entrenamiento

5.2.4. Accesibilidad

La accesibilidad es un aspecto esencial en el diseño de una página web, ya que garantiza que todos los usuarios, independientemente de sus habilidades o limitaciones, puedan acceder al contenido de manera efectiva. Por ello, se ha diseñado la aplicación web teniendo muy presentes las pautas de accesibilidad establecidas por la Web Content Accessibility Guidelines (WCAG)[8], un conjunto de directrices internacionales que proporcionan criterios y técnicas para garantizar que los sitios web sean accesibles para todos los usuarios, incluyendo aquellos con discapacidades.

Gestión de preferencias

Se ha implementado una sección de ajustes personalizables de forma que el usuario pueda adaptar la disposición de la aplicación web según sus necesidades y preferencias. Estos ajustes incluyen opciones como el tamaño de la fuente, el tipo de fuente, el interlineado, el espaciado del texto y la apariencia del cursor.

La capacidad de personalizar estos aspectos permite a cada usuario crear una experiencia única y accesible, especialmente para aquellos con dificultades de lectura o problemas visuales. Además, estos ajustes personalizables pueden mejorar la usabilidad general del sitio web y aumentar la satisfacción del usuario, ya que les permite adaptar el contenido a sus necesidades individuales.

Tamaño de la fuente

El usuario puede escoger entre cinco tamaños de fuente distintos. El tamaño predeterminado corresponde al nivel 3 y tiene la opción de aumentarlo o disminuirlo dos niveles más, respectivamente. De esta forma podrá adaptar el texto de forma personal y en función del dispositivo con el que esté accediendo al contenido. Los niveles que se ofrecen, ordenados de mayor a menor tamaño son:

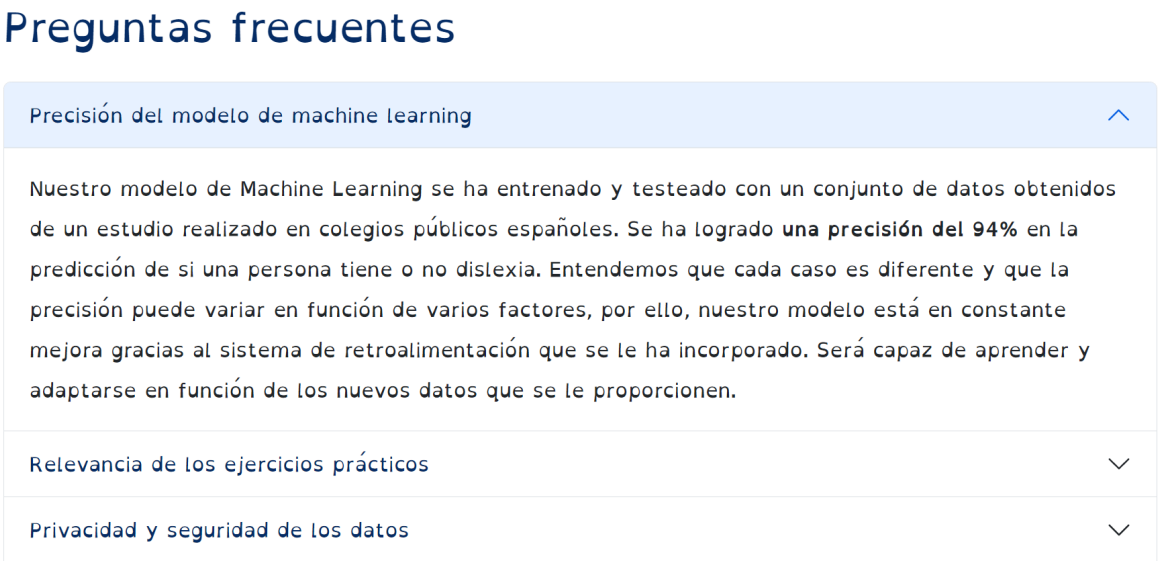


Figura 5.43: Tamaño de fuente 5

[8]https://www.w3.org/WAI/standards-guidelines/wcag/es

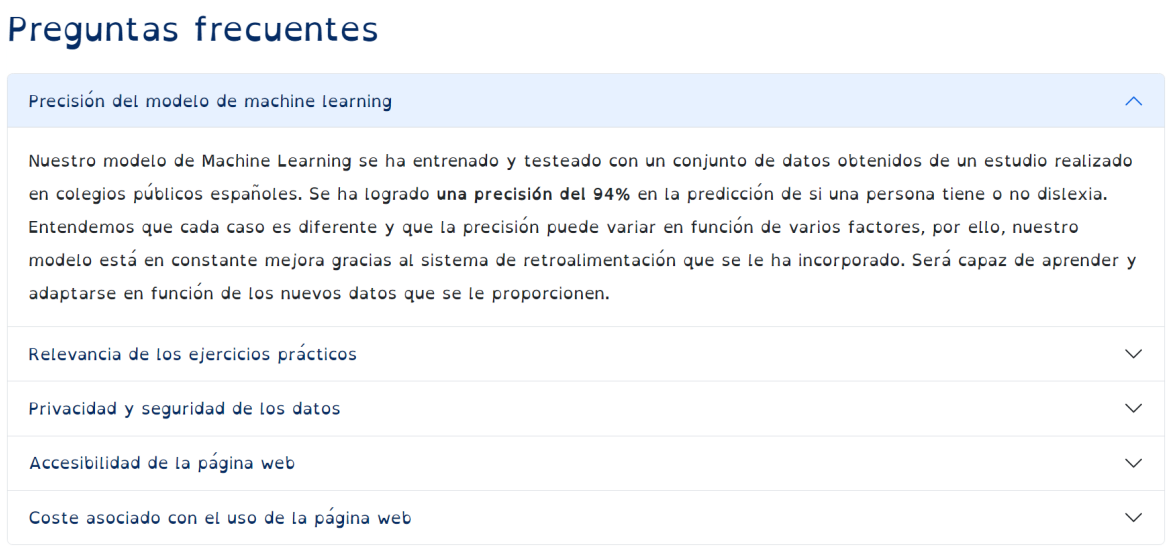


Figura 5.44: Tamaño de fuente 4

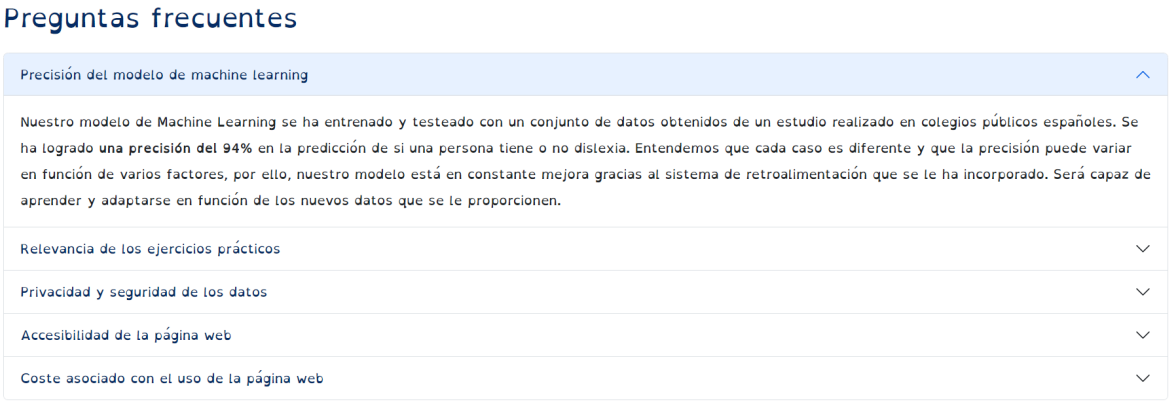


Figura 5.45: Tamaño de fuente 3

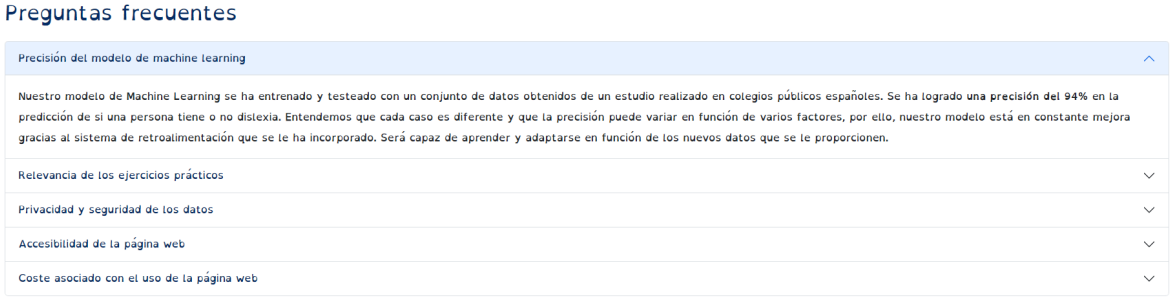


Figura 5.46: Tamaño de fuente 2

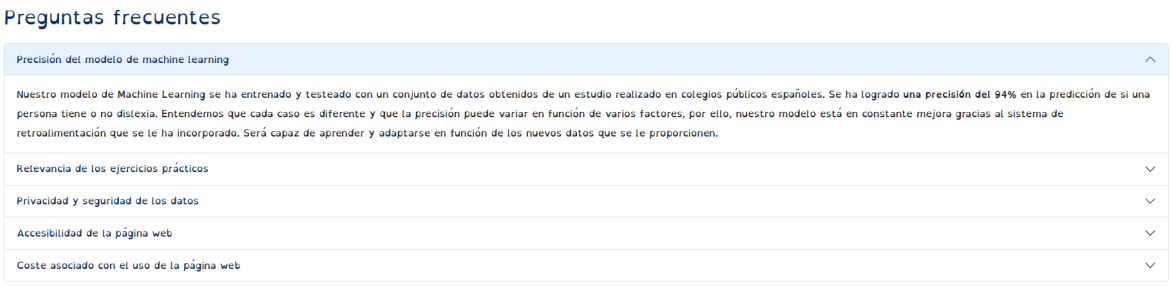


Figura 5.47: Tamaño de fuente 1

Tipo fuente

Se ha establecido que la fuente predeterminada sea Open Dyslexic, como se lleva mencionando hasta el momento y se muestra en la Figura 5.45, ya que es una fuente especialmente diseñada para mitigar algunos de los errores de lectura causados por la dislexia.

No obstante, para aquellos usuarios que les resulte más complicado leer el contenido ofrecido mediante este estilo de fuente, se ofrecen cuatro opciones de fuentes adicionales actuales y modernas con las que se puedan sentir más cómodos.

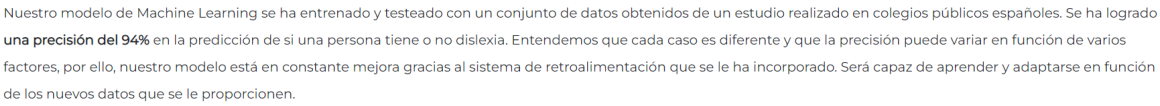


Figura 5.48: Fuente Monserrat

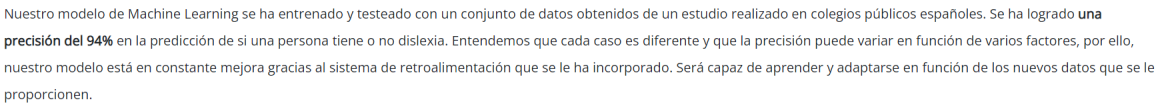


Figura 5.49: Fuente Open Sans

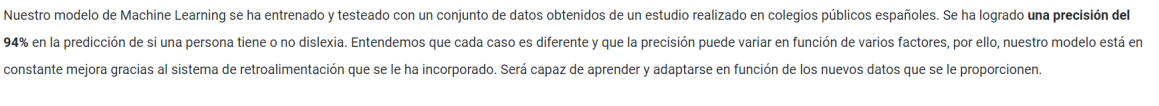


Figura 5.50: Fuente Roboto

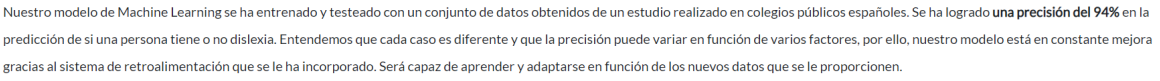


Figura 5.51: Fuente Lato

Interlineado

Se ofrece la posibilidad de modificar el interlineado del texto a elección del usuario. Al permitir a los usuarios ajustar el interlineado, se facilita la lectura y la comprensión del contenido, especialmente para aquellos con problemas de visión o dificultades de lectura como la dislexia.

Un interlineado adecuado puede mejorar la legibilidad al evitar que las líneas de texto se superpongan o se mezclen visualmente. El sitio web cuenta con tres niveles, siendo el predeterminado el segundo. Se puede aumentar o disminuir en función de las preferencias del usuario.

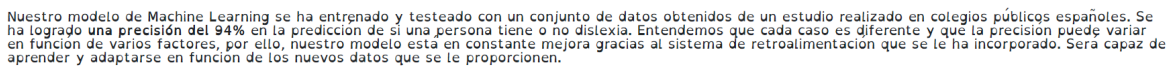


Figura 5.52: Nivel de interlineado 1

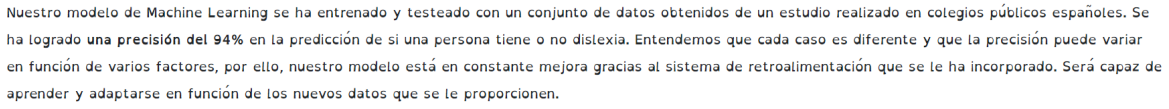


Figura 5.53: Nivel de interlineado 2

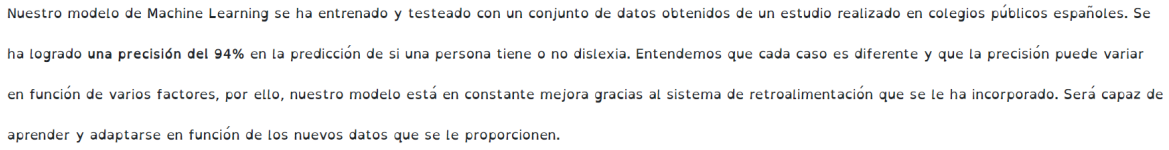


Figura 5.54: Nivel de interlineado 3

Espaciado del texto

El espaciado de texto se refiere a la distancia entre caracteres, palabras o párrafos. Se va a permitir a los usuarios ajustar el espaciado de texto, hasta dos niveles superiores al predeterminado (200 % y 400 %). Esto puede mejorar la legibilidad y facilitar la lectura para personas con dificultades visuales o de lectura.

Un espaciado adecuado puede prevenir la agrupación visual de caracteres o palabras y facilitar el seguimiento del texto en la pantalla. A continuación, se muestran las distintas opciones que ofrece la aplicación web:

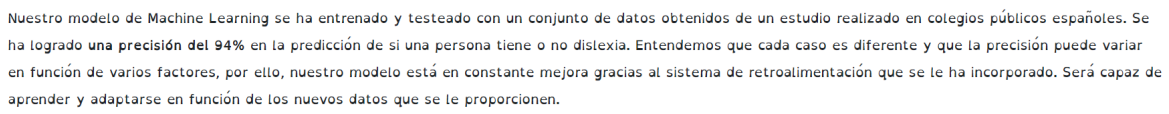


Figura 5.55: Espaciado predeterminado

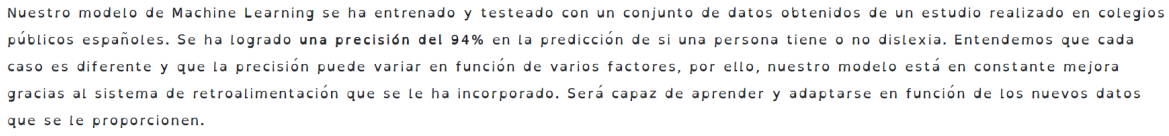


Figura 5.56: Espaciado 200 % superior

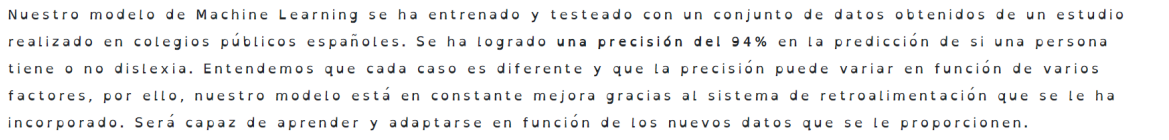


Figura 5.57: Espaciado 400 % superior

Cursor

El cursor es un indicador visual que muestra la posición actual del ratón en la pantalla. La aplicación web cuenta con hasta cuatro cursores distintos:

- El cursor predeterminado por el navegador web.
- Un cursor claramente visible, como el que se muestra en la Figura 5.58
- Un cursor con forma circular sin relleno
- Un cursor con forma cuadrada sin relleno

Permitir a los usuarios personalizar el cursor, puede mejorar la accesibilidad al facilitar su identificación y seguimiento. Puede resultar especialmente útil para personas con baja visión o problemas de percepción visual, ya que les ayuda a ubicar y seguir el cursor mientras interactúan con el contenido.

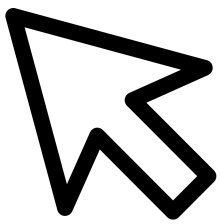


Figura 5.58: Cursor

Lector de pantalla

El sitio web ha sido diseñado para funcionar correctamente con lectores de pantalla, facilitando el acceso a la información para aquellos usuarios con discapacidades visuales o dificultades de lectura.

La herramienta que se ha empleado para testear si la aplicación web es compatible con tecnologías de asistencia ha sido el lector de pantalla ***NVDA*** (NonVisual Desktop Access). Se trata de un software gratuito y de código abierto que permite a personas con discapacidad visual acceder e interactuar las aplicaciones web.

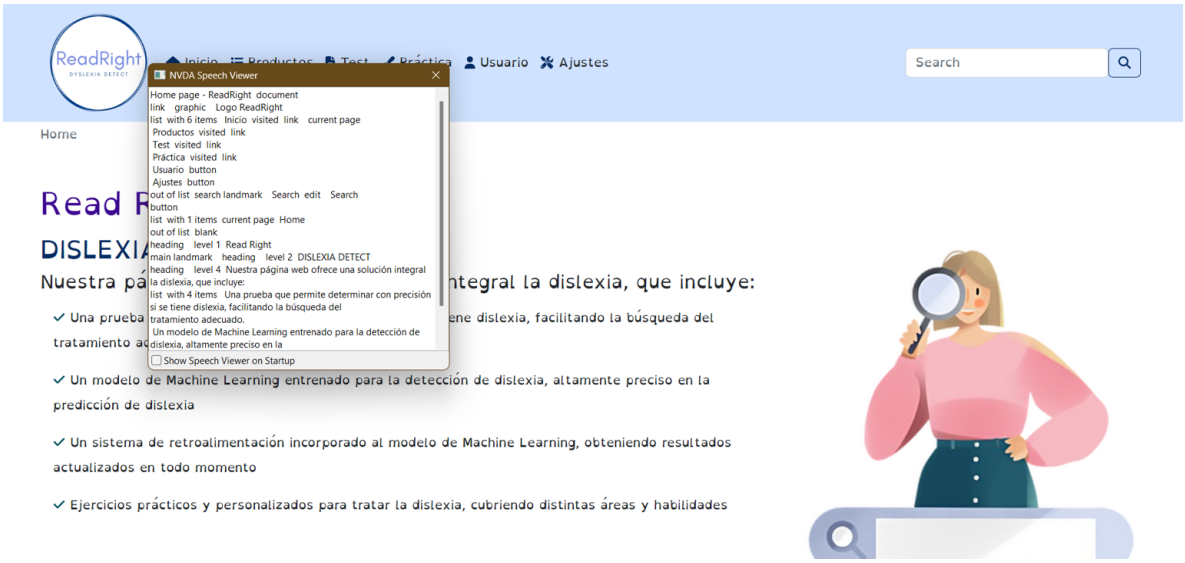


Figura 5.59: Página de Inicio usando un Lector de Pantalla

Movilidad con el Teclado

Se ha garantizado una navegación fluida y coherente mediante el teclado, permitiendo a los usuarios con limitaciones de movilidad o que no puedan utilizar un ratón interactuar fácilmente con el sitio web.

Figura 5.60: Acceso a "Productos" con el teclado

Contraste de Color

En base a la paleta de colores mencionada, se ha cuidado que el contraste entre los distintos elementos, ya sean texto o no, sea adecuado asegurando que el contenido sea legible para todos los usuarios, incluyendo aquellos con discapacidades visuales o daltonismo.

Se ha empleado la herramienta ***Colour Contrast Analyser*** para analizar que el color de primer plano y el color de fondo cumplan el contraste mínimo, ya sea texto o elementos no textuales, como tablas o subrayados. Esta herramienta cuida que las pautas 1.4.3 Contraste mínimo[9] y 1.4.11 Contraste sin texto de WCAG 2.1[10] sean aprobadas

Etiquetado adecuado

Todos los elementos interactivos y multimedia del sitio web están etiquetados apropiadamente, lo que facilita su identificación y comprensión por parte de los usuarios y las tecnologías de asistencia, como lectores de pantalla. Se han incluido encabezados y etiquetas ARIA para identificar correctamente el contenido que se ofrece.

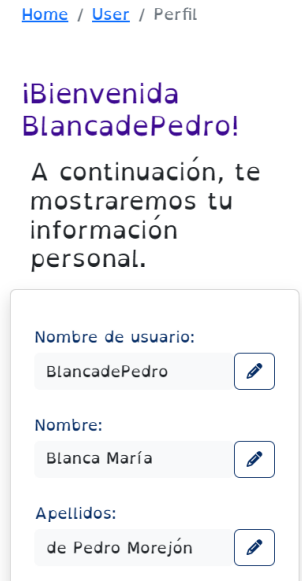
Además, se ha asegurado que la organización de los elementos sea coherente, siguiendo un patrón de lectura de izquierda a derecha y de arriba a abajo. De esta manera, el contenido ofrecido a las personas que requieren tecnologías asistidas o que no dispongan de un ratón brinda una experiencia similar a la de aquellos usuarios que no requieren adaptaciones adicionales.

Diseño Responsive

El diseño de la página web, como ya se ha mencionado en la sección que explica la experiencia de usuario, se adapta automáticamente a diferentes tamaños de pantalla y dispositivos, garantizando una experiencia de usuario óptima para todos, independientemente del dispositivo que utilicen para navegar por el sitio.

(a) Acceso al test

(b) Carrusel Página de Inicio



(c) Información de perfil

Figura 5.61: Dispositivo de 365x700

[9] https://www.w3.org/WAI/WCAG21/quickref/#contrast-minimum
[10] https://www.w3.org/WAI/WCAG21/quickref/#non-text-contrast

5.2.5. Experiencia de usuario

La experiencia de usuario (UX) es un aspecto fundamental en el diseño y desarrollo de una página web, ya que tiene un impacto directo en la satisfacción del usuario. Representa la forma en que los usuarios interactúan con el producto que se les ofrece a través del sitio web. Al proporcionar una experiencia de usuario óptima, se facilita la navegación y se mejora la interacción con el contenido proporcionado.

En el proceso de desarrollo y diseño del proyecto se ha tenido muy presente a quien se está dirigiendo el producto. El objetivo es ayudar a aquellas familias cuyos niños necesitan ayuda extra en el ámbito de la lectura. Por ello, los colores principales que se van a repetir a lo largo de la página son:

- **Azul**: Transmite **tranquilidad** y **confianza**, lo cual es esencial ya que se pretende que la plataforma sirva como una herramienta útil y confiable para aquellos que se encuentren en la situación de tener que usarla. A su vez, el azul evoca **profesionalismo**, sensación que es fundamental transmitir cuando se está trabajando con la salud de una persona.
- **Índigo**: Transmite **serenidad** e **inspiración**. El objetivo es que el usuario se sienta cómodo navegando por la página web en todo momento.
- **Morado**: Transmite **riqueza** junto a las sensaciones que ya transmitía el índigo, para enfatizar en la serenidad y tranquilidad que se quiere transmitir al usuario. El morado también puede sugerir una **experiencia única y personalizada**, lo que puede aumentar la satisfacción del usuario en la plataforma.

Tras seleccionar la paleta de colores se diseñó el logo de la aplicación web:

Figura 5.62: Logo ReadRight

Este diseño se hará presente a lo largo de la navegación web, variando ligeramente los tres elementos que lo componen: el icono de la bombilla, la línea separadora y el texto. Se mantendrá en todo momento el esquema de tonalidades, consiguiendo consistencia a lo largo del sistema de diseño. A continuación, se van a exponer los distintos puntos clave que conforman la experiencia de usuario.

Navegación y estructura web

La navegación y estructura del sitio web son cruciales para proporcionar una experiencia de usuario fluida y eficiente. La aplicación web presenta una estructura jerárquica clara y una navegación intuitiva, lo que permite a los usuarios encontrar fácilmente la información que buscan y moverse entre las diferentes secciones de la web fácilmente.

Para lograr que la navegación por la página web fuese intuitiva y fluida se han incluido diferentes elementos de diseño web que permiten al usuario saber en todo momento dónde se localizan, así como poder moverse con facilidad hacia cualquier sección o retroceder a la página de inicio. A su vez, se ha verificado que los enlaces son fácilmente reconocibles y accesibles.

Menú de navegación

Gracias a la herencia de Django el menú de navegación permanece constante en la cabecera, de forma que el usuario se pueda mover libremente por la página web. No se mostrará mientras se realiza el test de detección de dislexia, ni mientras se realizan los ejercicios de entrenamiento para que el usuario permanezca en la actividad que se encuentra hasta que la termine.

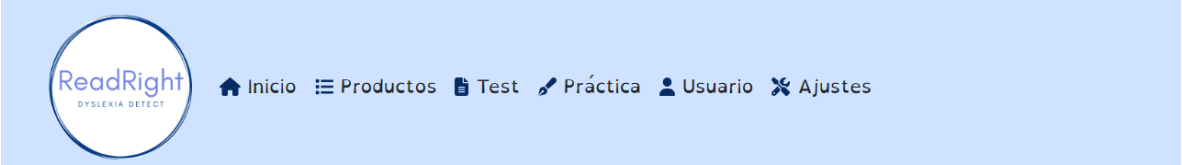


Figura 5.63: Menú de navegación

Migas de pan

Las migas de pan permiten a los usuarios identificar el recorrido que han realizado, con el objetivo de comprender dónde se encuentran dentro de la estructura del sitio web.

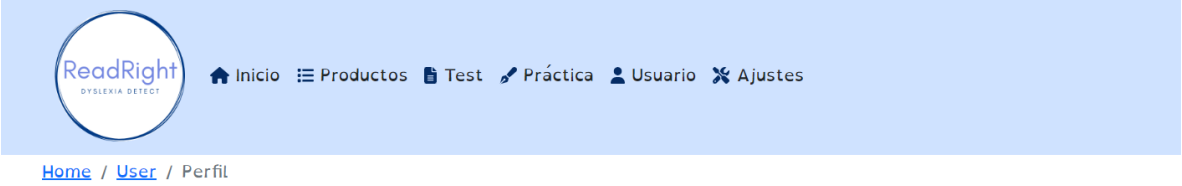


Figura 5.64: Migas de pan

En general, las migas de pan permiten a los usuarios regresar fácilmente a páginas anteriores. Sin embargo, en ciertos momentos, es posible que no se desee que el usuario regrese a la página de inicio a menos que sea estrictamente necesario. Por lo tanto, se han incluido las migas de pan con la navegación deshabilitada. De esta manera, el usuario será consciente de su ubicación en todo momento, pero no podrá retroceder hasta que haya completado la actividad que está realizando. En caso de que sea indispensable abandonar del test, se ha proporcionado la opción de salir presionando el botón situado arriba a la derecha.

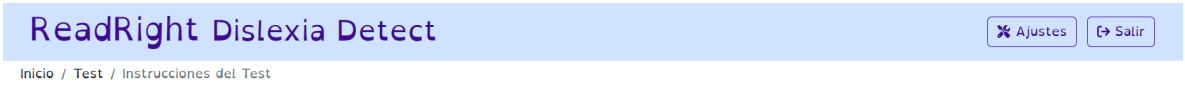


Figura 5.65: Migas de pan dentro del test de detección de dislexia

Mapa web

En el pie de página, se ha incluido un mapa web o *sitemap* para ayudar a los usuarios a identificar la estructura jerárquica y organización de las páginas del sitio web. Se ha optado por organizarlo de forma vertical y ubicarlo en la columna central, con el objetivo de captar la atención del usuario y permitirle acceder fácilmente a la sección que desee.

Cabe destacar que el pie de página también cuenta con otras dos columnas: la primera contiene información de contacto relevante, mientras que la tercera ofrece la opción de enviar comentarios.

Figura 5.66: Pie de página

Barra de Progreso

La barra de progreso se ha usado especialmente en el flujo del test de detección de dislexia para proporcionar al usuario información visual sobre cuanto ha realizado y lo que falta por completar.

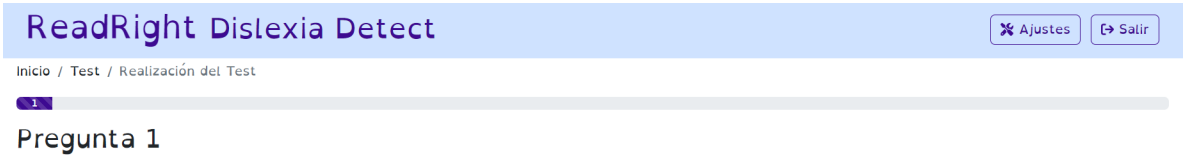


Figura 5.67: Enunciado de la Pregunta 1 del test

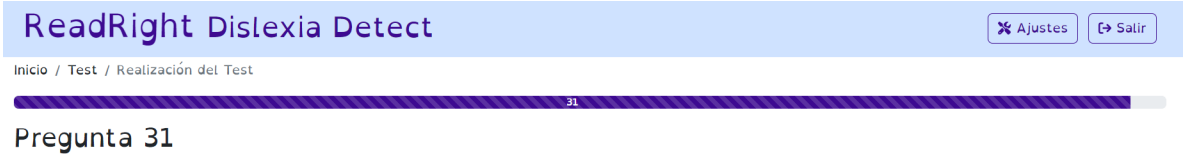


Figura 5.68: Enunciado de la Pregunta 31 del test

Diseño web adaptable

La aplicación web se ha diseñado de forma que sea capaz de adaptarse a los diferentes tamaños de pantalla. Esto va a permitir que los usuarios tengan acceso a todo el contenido web independientemente del dispositivo con el que estén navegando.

Se ha utilizado el *inspector* del navegador web para comprobar que el contenido se mantenga constante y que su disposición se adapte a la pantalla en todo momento. Además, se han incluido elementos familiares para el usuario en caso de que necesite modificar secciones de una vista. Por ejemplo, la disposición del menú de navegación varía cuando el ancho de la pantalla es menor de 990 píxeles, convirtiéndose en u menú de hamburguesa desplegable.

(a) Menú de navegación (b) Información (c) Pie de página

Figura 5.69: Dispositivo de 365x700

Diseño visual y estética

El diseño visual y la estética son esenciales para causar una buena primera impresión e invitar a los usuarios a explorar el contenido que se les ofrece. Como ya se ha mencionado, los colores predominantes en la página son el azul, el índigo y el morado, ya que evocan sensaciones de tranquilidad y confianza.

Además, se han incluido ilustraciones con el objetivo de crear un diseño atractivo y proporcionar apoyo visual al texto. Las ilustraciones mantienen la misma estética en todo momento, generando orden y coherencia en la página web.

La tipografía escogida es OpenDyslexic, diseñada específicamente para mitigar algunos de los errores de lectura causados por la dislexia[11]. No obstante, en la opción de ajustes, se ofrece una variedad de fuentes como Montserrat, Open Sans, Lato o Roboto, en caso de que el usuario desee modificarla.

[11] https://opendyslexic.org/

Interacción y retroalimentación

La interacción y retroalimentación son aspectos cruciales para garantizar una experiencia de usuario óptima en la página web. Se ha prestado especial atención a los formularios y otros elementos interactivos para asegurar una experiencia fluida y agradable.

En los formularios, se ha implementado una retroalimentación clara para ayudar a los usuarios a completarlos correctamente. Esto incluye indicadores visuales de campos obligatorios, mensajes de error informativos y validación en tiempo real para minimizar errores y guiar a los usuarios mientras completan los campos requeridos.

Figura 5.70: Registro del usuario en la página web

Además, se ha verificado que los elementos clickables reaccionen al interactuar con el ratón, proporcionando una retroalimentación visual que indica que el elemento es interactivo. Esto incluye cambios de color o de estilo al pasar el cursor sobre los botones, enlaces y otros elementos interactivos.

Los enlaces en el contenido del sitio web están claramente identificados al estar subrayados, lo que facilita la identificación y navegación por parte de los usuarios.

Contenido y legibilidad

Con el objetivo de garantizar que el contenido de la aplicación web sea fácil de leer y comprender, se ha prestado especial atención en estructuración del mismo.

En primer lugar, todas las vistas cuentan con un título principal capaz de describir el contenido mediante un par de palabras, seguido de una corta explicación que ponga en contexto al usuario. Se consigue facilitar la navegación y permiten a los usuarios identificar rápidamente los temas principales y encontrar la información que buscan.

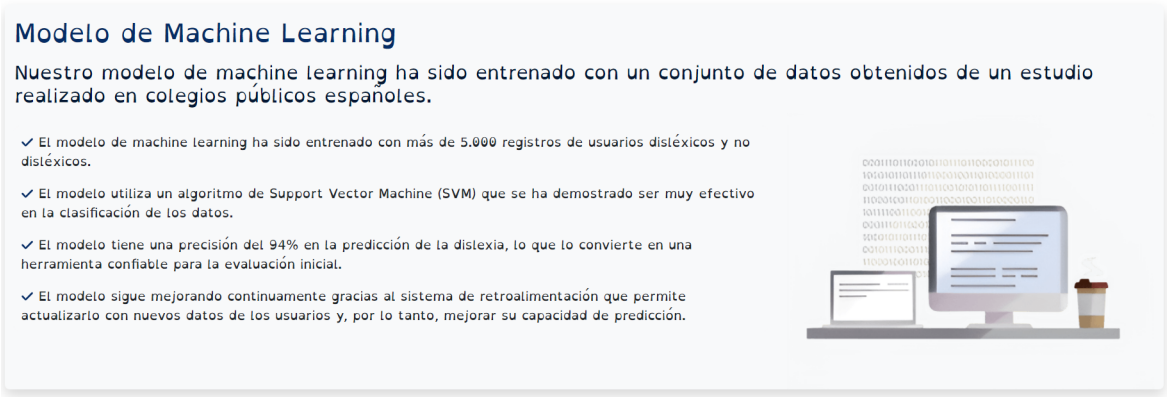


Figura 5.71: Información modelo de Machine Learning

Además, se ha organizado el texto en párrafos cortos y legibles, evitando bloques de texto densos que puedan dificultar la lectura. Esto permite a los usuarios leer y asimilar la información de manera más eficiente, lo que mejora la experiencia general del sitio web.

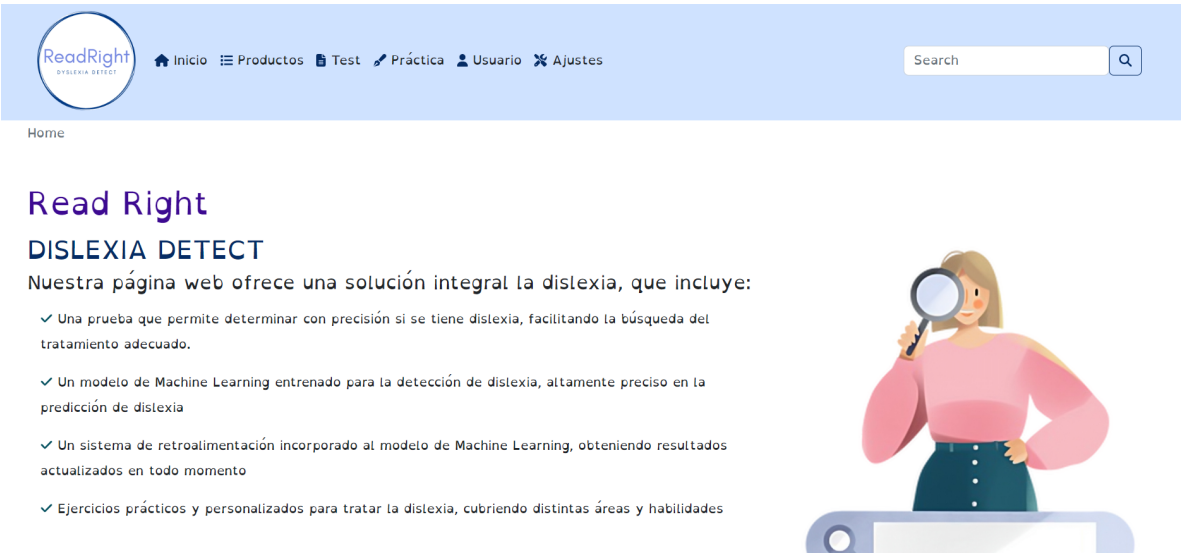


Figura 5.72: Página principal

Personalización y adaptabilidad

Se han implementado diversas características y funcionalidades en el sitio web para permitir a los usuarios personalizar su experiencia a sus necesidades. En la pestaña de *Ajustes* se despliega una serie de opciones de configuración que permiten al usuario modificar el interlineado y el espaciado de las letras, elegir la fuente con la que se sienta más cómodo, así como su tamaño y cambiar la forma del cursor aumentando su tamaño o sustituyéndola por otra figura.

5.3. Integración del modelo de Machine Learning en la aplicación web

En este capítulo, se describirá el proceso realizado para llevar a cabo la integración del modelo de Machine Learning de detección de dislexia en la aplicación web desarrollada con Django. Ya se ha expuesto la etapa del desarrollo del modelo de Machine Learning y la etapa de diseño y creación de la aplicación web.

A continuación, se explicarán los pasos realizados para integrar ambos procesos de forma fluida, consiguiendo que el usuario obtenga el resultado esperado tras la realización de la prueba de detección de dislexia.

5.3.1. Diseño de la Base de Datos

Inicialmente, se diseñó la base de datos con el objetivo de asociar un único test a un usuario, de manera que este solo pudiera realizarlo una vez. En el momento en que el usuario acceda a la prueba, se relacionará un objeto Test con el objeto Profile que haya iniciado sesión. Si un usuario no registrado intenta acceder al test, la página web le redirigirá a la vista de inicio de sesión para que se autentique.

De cada ejercicio de la prueba, es necesario almacenar el número de aciertos, fallos y clicks. El objetivo era introducir estos parámetros de forma dinámica; es decir, a medida que el usuario realice una actividad del test, se introducirán los valores mencionados en la base de datos. Por ello, se han definido tres objetos de tipo diccionario: *hits, misses y clicks*, los cuales se irán rellenando al final de cada ejercicio. Al finalizar el test, se obtendrán 32 elementos clave-valor con los resultados necesarios para trabajar con el modelo de Machine Learning.

```
from django.db import models
from register.models import Profile

class Test(models.Model):
    user_profile = models.OneToOneField(Profile, on_delete=models.CASCADE,
        unique=True)
    hits = models.JSONField(default=dict)
    misses = models.JSONField(default=dict)
    clicks = models.JSONField(default=dict)
    result = models.PositiveSmallIntegerField(null=True)
```

Listing 5.13: Clase *Test* de models.py

```
@login_required(login_url="/login")
def save_test_results(request, user_profile_id, index):
    user_profile = Profile.objects.get(pk=user_profile_id)
    test, _ = Test.objects.get_or_create(user_profile=user_profile)

    if request.method == 'POST':
        data = json.loads(request.body)
        hits = data['hits']
```

```
        misses = data['misses']
        clicks = data['clicks']

        test.hits[f'hits{index}'] = int(hits[index - 1])
        test.misses[f'misses{index}'] = int(misses[index - 1])
        test.clicks[f'clicks{index}'] = int(clicks[index - 1])
        test.save()

        return JsonResponse({'success': True})
    else:
        return JsonResponse({'success': False})
```

Listing 5.14: Clase*save_test_results* de views.py

Cada vez que el sistema llama a la función *save_test_results* se almacenarán los parámetros del índice correspondiente en la base de datos. Tras guardarlos se mostrará un mensaje indicando si la operación ha sido exitosa o no.

5.3.2. Importación del modelo de Machine Learning

Con el objetivo de cargar el modelo de Machine Laerning, se ha utilizado la librería de Python **pickle**, para exportar el modelo desde *ml_model.py*, situado en el directorio *data and ML model* e importándolo en el archivo *views.py* de la aplicación *prueba*.

Pickle[12] es una librería utilizada para serializar y deserializar objetos en Python. La serialización consiste en el proceso de convertir un objeto en una cadena de bytes que puede ser almacenada en un archivo. Esta cadena se puede transmitir o guardar en una base de datos para su posterior uso. La deserialización se trata del proceso inverso, convirtiendo la cadena de bytes de vuelta en un objeto Python.

```
##Support Vector Machines
from sklearn import svm
# Train Model and Predict
mod_SVM = svm.SVC(kernel='poly', degree=15, coef0=7).fit(X_train, y_train)
# Test Model
y_hat_SVM = mod_SVM.predict(X_test)

#save the model
pickle.dump(mod_SVM, open("ml_model.sav", "wb"))
```

Listing 5.15: Importación del modelo de Machine Learning

```
import pickle

@login_required(login_url="/login")
def results(request, user_profile_id):
    #ML model
```

[12]https://docs.python.org/3/library/pickle.html

```
    model = pickle.load(open('ml_model.sav', 'rb'))
```

Listing 5.16: Exportación del modelo de Machine Learning

5.3.3. Base de Datos

Una vez que el modelo está importado en la aplicación web, solo queda acceder a los datos almacenados en la base de datos, reorganizarlos de acuerdo con las necesidades del modelo y transformarlos en un objeto DataFrame para poder realizar la predicción y mostrar el resultado en pantalla.

Para ello se va a completar la función *results()* mostrada anteriormente, para que calcule la predicción del usuario y almacene el resultado en la base de datos. De esta forma el usuario podrá acceder a los resultados obtenidos de la prueba rápidamente, ya que solo será necesario acceder a la información almacenada en la base de datos:

```
@login_required(login_url="/login")
def results(request, user_profile_id):
    #ML model
    model = pickle.load(open('ml_model.sav', 'rb'))

    #DDBB
    test = Test.objects.get(user_profile_id=user_profile_id)
    user = Profile.objects.get(pk=user_profile_id)

    #Data
    hits = test.hits
    misses = test.misses
    clicks = test.clicks
    #Gender = user.sexo
    Gender = 0 if user.sexo == "Hombre" else 1
    #Nativelang = user.lengua
    Nativelang = 0 if user.lengua == "No" else 1
    #Otherlang = user.suspender
    Otherlang = 0 if user.suspender == "No" else 1
    Age = user.edad

    #user info
    test_data = {"Gender": Gender, "Nativelang": Nativelang, "Otherlang":
        Otherlang, "Age": Age}

    #test info
    for i in range(1, 33):  # Itera desde 1 hasta 32
        clicks_i = clicks.get(f"clicks{i}", None)
        hits_i = hits.get(f"hits{i}", None)
        misses_i = misses.get(f"misses{i}", None)

        # add data
        test_data[f"Clicks{i}"] = clicks_i
        test_data[f"Hits{i}"] = hits_i
        test_data[f"Misses{i}"] = misses_i

```

```
36      #data ML model
37      df = pd.DataFrame([test_data])
38      print(df)
39
40      #predict
41      prediction = model.predict(df)
42      print("Eres dis xico: ",prediction)
43
44      #save prediction DDBB
45      test.result = 1 if prediction else 0
46      test.save()
47
48      return render(request, 'prueba/results.html', {'prediction': int(
            prediction), 'Data': test_data})
```

Listing 5.17: Clase *results* de views.py

Finalmente, tras calcular la predicción sobre si el usuario tiene dislexia o no mediante el modelo de Machine Learning, se redirigirá al usuario a una nueva vista que le informará del resultado. La vista contará con un encabezado respondiendo a la pregunta de si es disléxico y seguido se le mostrará una tabla con la puntuación obtenido en cada actividad, así como los procesos cognitivos asociados a cada ejercicio.

Capítulo 6

Análisis de resultados

En el siguiente capítulo se van a analizar los resultados obtenidos del proyecto realizado, tras implementar las funcionalidades expuestas en el Capítulo 5.

Tal y como se expuso en la Sección 4.1, la intención con la que se decidió desarrollar el presente Trabajo Fin de Grado era diseñar un modelo de Machine Learning capaz de predecir con el mínimo porcentaje de error si una persona tiene o no dislexia, consiguiendo proponer una alternativa a los métodos de predicción y detección tradicionales de dicho trastorno, los cuales no siempre son correctos ni precisos.

Junto con el modelo de Machine Learning, se ha querido diseñar el test en el que se basa el estudio del que se recogen los datos con los que se ha entrenado y evaluado el modelo, proporcionando una herramienta *online* accesible y orientada a ayudar a las familias que lo necesitan.

En dicha aplicación no solo se ha incluido la prueba de detección de dislexia, sino que también ejercicios de entrenamiento para practicar en un entorno dinámico y familiar aquellas habilidades que se han visto comprometidas por el trastorno.

La aplicación web cuenta con información detallada sobre el modelo mencionado, las funcionalidades que ofrece y la forma en la que se ha conseguido diseñar un entorno accesible y personalizable para cualquier usuario, independientemente de la condición en la que se encuentre.

Se han realizado numerosas pruebas con el fin de verificar el correcto funcionamiento de todos los componentes que conforman la aplicación y que las funcionalidades que se han integrado actúen como se espera.

A continuación, se van a analizar los resultados obtenidos del proyecto desarrollado y verificar el correcto comportamiento del mismo.

6.1. Análisis del modelo de Machine Learning

Se va a proporcionar un análisis en profundidad del modelo de Machine Learning elegido en la Sección 5.1. Se discutirán los motivos de la elección de este modelo, exponiendo sus ventajas y desventajas, así como su rendimiento y precisión en la tarea de detectar la dislexia.

Como se mencionó en el Capítulo 5, para el diseño del modelo de detección de dislexia se disponía de dos conjuntos de datos distintos: *Desktop* que inicialmente se destinó para el entrenamiento y *Dataset* para el testeo. A diferencia del primero, el segundo presentaba valores nulos, los cuales se gestionaron mediante distintas técnicas, lo que resultó en plantear tres escenarios distintos:

- **Esceario 1**: Empleo de ambos *dataset*, *Desktop* para el entrenamiento y *Tablet* para el testeo.
- **Escenario 2**: Uso único del *dataset Desktop*.
- **Escenario 3**: Uso único del dataset *Tablet*.

A su vez, fue necesario implementar diversas técnicas de balanceo de desproporción de clases, ya que la cantidad de casos negativos era muy superior a los casos positivos. Se llevó a cabo la implementación de tres técnicas distintas con el fin de determinar cuál se adaptaba mejor a los conjuntos de datos: **Undersampling**[1], **Oversampling**[2] y **SMOTE**[3].

Finalmente, debido a la naturaleza del problema, se seleccionaron cinco algoritmos distintos de aprendizaje supervisado: **K-Nearest Neighbors, Logistic Regression, Support Vector Machine, Decision Tree y Random Forest**. Independientemente del algoritmo elegido, el modelo se entrena utilizando datos etiquetados (*Dyslexia*: 0 o 1). Esto ha permitido realizar predicciones basadas en los datos de entrada, lo cual facilitará la clasificación de las muestras y la obtención del modelo esperado.

Tal y como se mostró al final de la primera parte del Capítulo 5, en la Sección 5.1.8, el algoritmo elegido ha sido **Support Vector Machine** alimentado únicamente mediante el conjunto de datos *Desktop* aplicando la técnica de balanceo de desproporción de clases **Oversampling**.

Aunque se obtuvieron resultados muy diversos, debido a la cantidad de alternativas de las que se disponía, existen diversas razones por las que se ha considerado que esta es la óptima y la que mejor capacidad de predicción presenta, teniendo en cuenta todas las métricas de evaluación mencionadas en la Sección 3.3. A continuación, se van a justificar las decisiones tomadas, exponiendo los modelos finalistas y sus características.

6.1.1. SMOTE

La técnica de *SMOTE* permite la interpolación de muestras en el conjunto de datos existente, lo que conduce a la generación de nuevas instancias. Como consecuencia, las dimensiones del *dataset* han experimentado un notable incremento.

[1]Eliminar aleatoriamente muestras de la clase mayoritaria hasta lograr el equilibrio con la clase minoritaria.
[2]Agregar aleatoriamente muestras de la clase minoritaria hasta lograr el equilibrio con la clase mayoritaria.
[3]Interpolación de los casos existentes con el objetivo de crear nuevas instancias de la clase minoritaria hasta lograr el equilibrio.

Analizando los modelos obtenidos la implementación de cada algoritmo en los diferentes escenarios, se ha observado que el modelo que mejores resultados ofrece se obtiene a partir del algoritmo Support Vector Machines entrenado con los datos del *dataset Desktop*:

Support Vector Machines	*Accuracy*	*Precision*	*Recall*	*F1-Score*
0	0.95	1.00	0.90	0.95
1		0.91	1.00	0.95

Cuadro 6.1: Evaluación de los resultados del algoritmo Support Vector Machines empleando el dataset Desktop

Se puede observar que los resultados obtenidos son excelentes, lo que indica un modelo con un rendimiento destacado. Teniendo presente que este podría corresponder al modelo finalista que se implemente en la aplicación web, es necesario analizar los modelos proporcionados tras aplicar la técnica *Oversampling* también.

6.1.2. Oversampling

La técnica de *Oversampling* consiste en replicar muestras aleatoriamente de la clase minoritaria con el objetivo de balancear los datos. Como consecuencia, las dimensiones del *dataset* incrementan.

Esta técnica se ha comportado positivamente en los escenarios donde solo se utilizaba un conjunto de datos, es decir, cuando se entrenaba el modelo o bien con el *dataset Desktop* o con el *dataset Tablet*. En el escenario en el que se utilizan ambos las dimensiones crecen considerablemente, impidiendo al modelo predecir las clases de forma eficiente.

Los valores resultantes de las medidas de evaluación del rendimiento del modelo no eran directamente comparables con los resultados obtenidos al entrenar el modelo utilizando solo uno de los *dataset* disponibles. Es por ello que los modelos obtenidos en el escenario 1 se descartaron. No obstante, sí se obtuvieron resultado prometedores en los escenarios 2 y 3.

K-Nearest Neighbors empleando el *dataset Desktop*

Implementando el algoritmo **K-Nearest Neighbors**, alimentado con los datos del *dataset Desktop* se obtiene:

K-Nearest Neighbors	*Accuracy*	*Precision*	*Recall*	*F1-Score*
0	0.95	1.00	0.88	0.93
1		0.92	1.0	0.96

Cuadro 6.2: Evaluación de los resultados del algoritmo K-Nearest Neighbors empleando el *dataset Desktop*

Support Vector Machines empleando el *dataset Desktop*

Implementando el algoritmo **Support Vector Machines**, alimentado con los datos del *dataset Desktop* se obtiene:

Support Vector Machines	*Accuracy*	*Precision*	*Recall*	*F1-Score*
0	0.96	1.00	0.9	0.94
1		0.93	1.0	0.97

Cuadro 6.3: Evaluación de los resultados del algoritmo Support Vector Machines empleando el *dataset Desktop*

Tree Decision empleando el *dataset Desktop*

Implementando el algoritmo **Tree Decision**, alimentado con los datos del *dataset Desktop* se obtiene:

Tree Decision	*Accuracy*	*Precision*	*Recall*	*F1-Score*
0	0.94	1.00	0.85	0.91
1		0.91	1.0	0.95

Cuadro 6.4: Evaluación de los resultados del algoritmo Tree Decision empleando el *dataset Desktop*

K-Nearest Neighbors empleando el *dataset Tablet*

Implementando el algoritmo **K-Nearest Neighbors**, alimentado con los datos del *dataset Tablet* se obtiene:

K-Nearest Neighbors	Drop				Zero				Mean				Median			
	Accuracy	*Precision*	*Recall*	*F1-Score*	*Accuracy*	*Precision*	*Recall*	*F1-Score*	*Accuracy*	*Precision*	*Recall*	*F1-Score*	*Accuracy*	*Precision*	*Recall*	*F1-Score*
0	0.96	0.95	0.97	0.96	0.96	1.00	0.89	0.94	0.94	0.95	0.89	0.92	0.94	0.95	0.89	0.92
1		0.97	0.94	0.96		0.93	1.00	0.96		0.93	0.97	0.95		0.93	0.97	0.95

Cuadro 6.5: Evaluación de los resultados del algoritmo K-Nearest Neighbors empleando el dataset Tablet

Tras analizar los resultados obtenidos de los diferentes algoritmos evaluados, se puede concluir que el modelo correspondiente a la Tabla 6.3 destaca por encima del resto en términos de rendimiento. Los valores obtenidos en las medidas de evaluación, como la exactitud (*Accuracy*), la precisión (*Precision*), el recuerdo (*Recall*) y la puntuación F1 (*F1-Score*), superan significativamente a los obtenidos por los demás algoritmos analizados. Estos resultados respaldan la elección del algoritmo *SVM* como la opción más efectiva para abordar el problema en cuestión. Finalmente, se va a mostrar la matriz de confusión calculada para el modelo elegido:

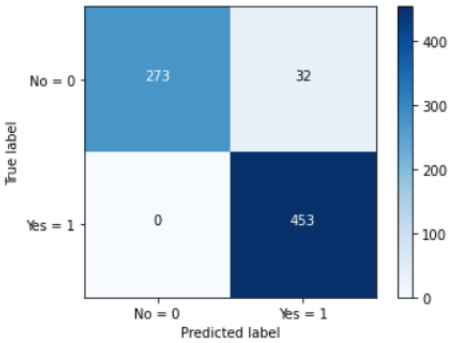


Figura 6.1: Matriz de confusión del modelo final

6.2. Análisis de los resultados proporcionados por la Aplicación Web

Se va a mostrar como se presentan los resultados generados por la aplicación web, enfocándose en cómo la aplicación interpreta y presenta los resultados del modelo de Machine Learning a los usuarios.

Tras realizar el test el usuario obtendrá los resultados de la predicción realizada por la interfaz de usuario, la cual se divide en dos zonas principales: un encabezado donde se especifica si la persona padece o no de dislexia, según los resultados proporcionados por el modelo de detección de dislexia, y una tabla compuesta por treinta y dos filas, cada una correspondiente a una pregunta distinta del test, especificando el proceso cognitivo que se trabaja en cada pregunta, así como las puntuaciones correspondientes. De esta forma, el usuario será capaz de identificar en que preguntas ha presentado mayor dificultad y podrá trabajar aquellas habilidades que considere necesarias.

Al tratarse de un problema de clasificación, el modelo de Machine Learning solo devuelve dos resultados: sí o no, en función de si una persona tiene dislexia o no, según su actuación en la prueba.

6.2.1. Escenario 1: Resultado positivo

Se presentan los resultados proporcionados por la aplicación web en casos donde se detecta dislexia:

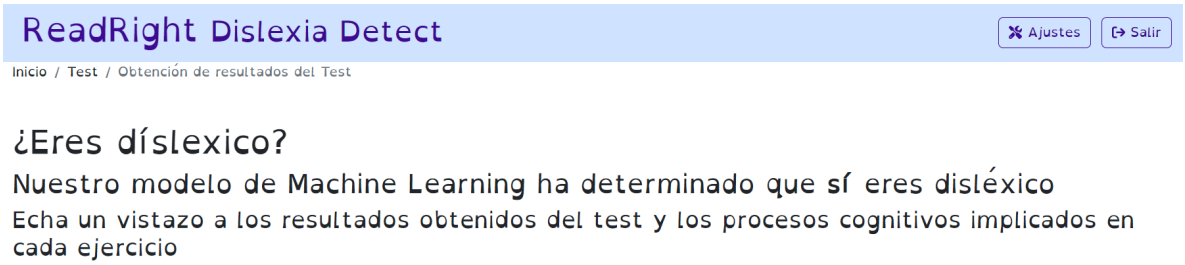


Figura 6.2: El usuario presenta dislexia

6.2.2. Escenario 2: Resultado negativo

Se exponen los resultados proporcionados por la aplicación web en casos donde no se detecta dislexia.

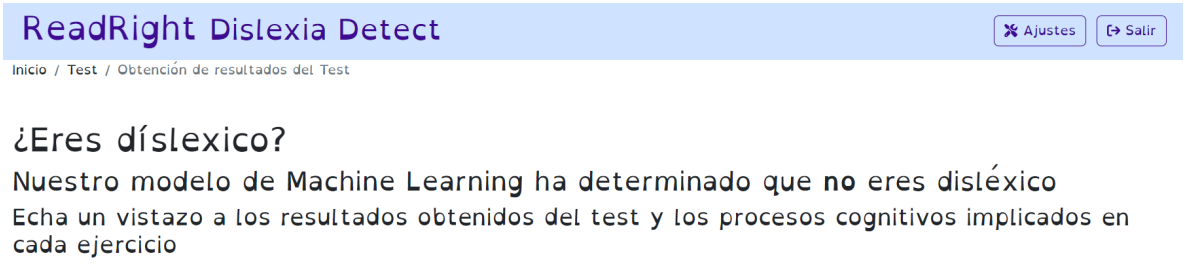


Figura 6.3: El usuario no presenta dislexia

6.3. Análisis de la Aplicación Web

Esta sección se centrará en evaluar el funcionamiento general de la aplicación web. Se discutirán aspectos como la usabilidad y la accesibilidad, así como la eficiencia y la estabilidad de la aplicación. Este análisis es crucial para garantizar que la aplicación web cumpla con sus objetivos.

6.3.1. Pruebas unitarias

Las pruebas unitarias que se han implementado son cruciales para asegurar el correcto funcionamiento de la aplicación web. Las pruebas cubren aspectos claves del sistema como la autenticación de usuarios, el registro, la actualización de información del perfil, el cambio de contraseña y la funcionalidad de la aplicación principal.

El objetivo de estas pruebas unitarias es validar que cada función clave de la aplicación funciona como se espera y contribuye al correcto rendimiento global del sistema. Las pruebas se han realizado durante el proceso de desarrollo para asegurar que las actualizaciones o cambios en el código no interrumpan ninguna funcionalidad existente.

Pruebas unitarias para la aplicación Prueba

La primera serie de pruebas se centra en el funcionamiento de la aplicación principal. Se va a verificar el correcto funcionamiento de la funcionalidad correspondiente a la prueba de detección de dislexia, desde el momento en el que el usuario decide acceder a ella hasta la presentación de los resultados finales, habiendo realizado todas las preguntas que conforman el test.

Se ha creado un usuario de prueba, verificando que cuando accede a la vista del inicio de la prueba se genera un objeto *Test* asociado a dicho usuario, en la base de datos, donde se van a almacenar los resultados de cada ejercicio.

Adicionalmente, se ha comprobado que los resultados del test se guardan de manera correcta en la base de datos tras cada *POST request*.

Pruebas unitarias para la aplicación Register

Se han realizado una serie de pruebas centradas en el manejo de los usuarios en la aplicación. Se han realizado pruebas de autenticación, creación de nuevos usuarios, actualización de los campos del perfil de usuario y eliminación de un usuario existente. A su vez, se incluye una verificación para comprobar que se puede actualizar la contraseña con éxito.

Finalmente, se ha establecido una prueba para el restablecimiento de la contraseña. En esta prueba, se envía un correo electrónico de restablecimiento de contraseña a la dirección de correo electrónico que el usuario introduzca y se confirma que el enlace de restablecimiento de la contraseña dirige al usuario a la página de confirmación de restablecimiento de contraseña. Luego, se verifica que el usuario puede introducir una nueva contraseña y acceder al sistema mediante dicha contraseña.

A partir de estos resultados, se puede concluir que la aplicación web está funcionando correctamente según las expectativas y se ha conseguido desarrollar una aplicación segura y eficiente.

Capítulo 7

Conclusiones y Trabajos Futuros

En este capítulo, se van a exponer las conclusiones y los resultados más relevantes obtenidos tras la realización de este proyecto. Se examinarán detalladamente los objetivos planteados inicialmente y se evaluarán los logros alcanzados a lo largo del proceso. Además, se discutirán diferentes trabajos futuros relacionados con el proyecto desarrollado, con el objetivo de continuar expandiéndolo.

7.1. Conclusiones y resultados principales

El objetivo principal que motivó el desarrollo del presente Trabajo Fin de Grado se ha logrado. Como se mostró en el Capítulo 6, se ha alcanzado dicho objetivo al haber diseñado un modelo de Machine Learning capaz de detectar la dislexia con una precisión del 96 %. Este modelo va a permitir diagnosticar la dislexia de forma rápida y eficaz.

Con respecto a los objetivos expuestos en la Sección 4.2 se puede verificar que:

- Se ha logrado diseñar el modelo de detección de dislexia e implementarlo en la aplicación web. El modelo desarrollado en Python se ha integrado en el *back-end* de forma fluida. El usuario simplemente debe completar la prueba que se le ofrece en la aplicación y obtendrá los resultados inmediatamente después de su finalización. El rendimiento del modelo se mantiene, permitiendo el acceso de esta herramienta al público.

- Se ha desarrollado una aplicación web funcional y completa, lista para su lanzamiento al público. Esta aplicación consta de dos funcionalidades principales: un test de detección de dislexia con un rendimiento excelente, y seis ejercicios de entrenamiento sencillos e intuitivos, diseñados específicamente para trabajar en áreas concretas. A su vez, se ha incluido información adicional con el objetivo de facilitar la navegación por la aplicación y proporcionar una experiencia de usuario óptima.

- Se ha diseñado una aplicación web con accesibilidad en mente, asegurando su compatibilidad con diversas herramientas externas. Se ha prestado especial atención en definir claramente los elementos utilizados, de manera que la información no solo sea accesible visualmente, sino también

mediante tecnologías asistivas[1]. Se han incluido descripciones de imágenes, se ha proporcionado un contraste adecuado de colores y se ha utilizado una estructura de navegación clara y consistente, garantizando que la aplicación pueda ser utilizada por personas con discapacidades visuales, auditivas o motrices.

Adicionalmente, la aplicación cuenta con un módulo de autenticación y gestión de cuentas completo y seguro. Los usuarios podrán crear una cuenta de forma sencilla y guiada, gracias a la implementación de mensajes aclaratorios que facilitan el proceso. Asimismo, se requiere autenticación cada vez que se accede a la aplicación para garantizar la seguridad de la cuenta. El módulo de gestión de cuentas permite a los usuarios realizar diversas acciones. Por ejemplo, pueden eliminar su cuenta si así lo desean.

Además, los usuarios tienen la capacidad de modificar su información personal, como nombre, edad o correo electrónico, permaneciendo actualizada en todo momento. De igual manera, tienen la posibilidad de cambiar su contraseña, lo que brinda un nivel adicional de seguridad.

En caso de que un usuario olvide su contraseña, la aplicación ofrece una función de recuperación de cuenta. Mediante la introducción del correo electrónico asociado a la cuenta, el usuario puede seguir un proceso para restablecer su contraseña y recuperar el acceso a su cuenta.

7.2. Trabajos Futuros

En esta sección, se van a exponer posibles adiciones al proyecto finalizado:

Sistema de retroalimentación del modelo de Machine Learning

Se propone implementar un sistema de retroalimentación para el modelo de Machine Learning, lo que permitiría su actualización en función de los nuevos datos recopilados desde la aplicación web. Esta retroalimentación permitiría al modelo adaptarse y mejorar su capacidad para predecir la dislexia con mayor precisión, al incorporar nuevas muestras al conjunto de datos utilizado para entrenar y evaluar el modelo. Estos datos provienen de la aplicación web, lo que garantiza que correspondan a información actualizada, manteniendo así el modelo al día y ofreciendo un rendimiento óptimo de manera continua.

El desarrollo de este sistema de retroalimentación requeriría la implementación de una estructura que permita recopilar y procesar los datos provenientes de la aplicación web de manera eficiente. Además, sería necesario establecer un mecanismo de actualización periódica del modelo, utilizando los nuevos datos recopilados para reentrenar y ajustar los parámetros del modelo existente.

Este enfoque de retroalimentación permitiría aprovechar la información obtenida en tiempo real a través de la aplicación web, lo cual resulta especialmente útil en el contexto de la detección y predicción de la dislexia. Al incorporar constantemente nuevos datos, el modelo estaría en condiciones de adaptarse a posibles cambios o variaciones en los patrones y características de la dislexia, mejorando su capacidad de diagnóstico y predicción de forma automática.

[1]Herramientas utilizadas por personas con dificultades para realizar tareas regulares con el objetivo de mejorar su calidad de vida.

Adaptación de la aplicación web a una aplicación móvil

La adaptación de la aplicación web existente a una aplicación móvil implicaría desarrollar una versión específica de la aplicación que sea compatible con dispositivos móviles, como *smartphones* y *tablets*.

La adaptación a una aplicación móvil tendría varias ventajas y beneficios. En primer lugar, permitiría a los usuarios acceder a la funcionalidad de la aplicación de manera más conveniente y portátil, ya que podrían utilizarla en cualquier momento y lugar desde sus dispositivos móviles. Esto aumentaría la accesibilidad y la comodidad de uso, lo que podría fomentar una mayor participación por parte de los usuarios.

Aunque la aplicación web se diseñó para poder adaptarse a distintos tamaños de pantalla, será necesario realizar ciertas modificaciones para considerar las interacciones táctiles características de los dispositivos móviles, como deslizar, tocar y hacer zoom, para garantizar una navegación intuitiva y cómoda para los usuarios. Se tendrían que ajustar los elementos interactivos, así como la disposición de los botones y elementos de navegación para facilitar su uso en pantallas táctiles.

Además, se tendría que considerar la compatibilidad con diferentes plataformas móviles, como iOS y Android, para llegar a una mayor audiencia de usuarios. Esto implicaría desarrollar versiones específicas de la aplicación para cada plataforma o explorar opciones de desarrollo multiplataforma.

Ampliación de la funcionalidad de los ejercicios de entrenamiento

En la actualidad, la aplicación proporciona seis opciones distintas de juegos y actividades recomendadas para tratar la dislexia. Sin embargo, se plantea la posibilidad de ampliar esta funcionalidad mediante la inclusión de nuevos ejercicios clasificados según las competencias específicas en las que se centren.

El objetivo es poner a disposición del público una herramienta interactiva e intuitiva que aborde de manera integral la dislexia. Esta ampliación de la funcionalidad permitiría abordar diferentes aspectos relacionados con el trastorno, como la lectura, la escritura, la discriminación visual y otras habilidades cognitivas afectadas.

Al incluir nuevos ejercicios, se busca brindar a los usuarios una mayor variedad de opciones y enfoques para trabajar las áreas específicas afectadas por la dislexia. Estos nuevos ejercicios podrían estar diseñados de manera cercana y atractiva, fomentando la motivación y el compromiso de los usuarios durante el proceso de entrenamiento.

Asimismo, la clasificación de los ejercicios según las competencias en las que se centren permitiría a los usuarios identificar y seleccionar los ejercicios más relevantes para sus necesidades individuales. Esto proporcionaría una experiencia más personalizada y adaptada a las dificultades particulares de cada usuario.

Incorporación de un *Dashboard* para el seguimiento del usuario

Hilando con la ampliación de la funcionalidad asociada al entrenamiento de la dislexia, se propone añadir una nueva funcionalidad que permita a los usuarios llevar un seguimiento detallado de su actividad y avances, incluyendo un cuadro de mando que refleje de manera visual la evolución de cada usuario.

Esta nueva funcionalidad permitirá a los usuarios tener un mayor control sobre su progreso en el tratamiento de la dislexia, lo que a su vez les motivará a continuar practicando y superando los desafíos asociados a este trastorno del aprendizaje. El cuadro de mando proporcionará una visión general de los logros y mejoras realizadas, brindando una representación gráfica del progreso individual.

Se incluirán gráficas que reflejen el progreso semanal y mensual, clasificando los avances en función de las competencias específicas que se trabajen. Esto permitirá a los usuarios identificar las áreas en las que necesitan más enfoque y práctica.

Además, el seguimiento del progreso no solo proporcionará información valiosa para los usuarios, sino que también será una herramienta útil para los profesionales de la salud y docentes que acompañan a los usuarios en su proceso de tratamiento. El *dashboard* les permitirá evaluar y ajustar las estrategias y enfoques utilizados en función de los resultados obtenidos, mejorando así la eficacia del tratamiento y ofreciendo una atención más personalizada.

Bibliografía

[1] Universia Fundación. 1 de cada 10 escolares padece dislexia. [En línea] Disponible en: https://www.universia.net/es/actualidad/orientacion-academica/1-cada-10-escolares-padece-dislexia-1115116.html, 2014. (Accedido: 29-05-2023).

[2] Dislexia. [En línea] Disponible en: https://www.mayoclinic.org/es-es/diseases-conditions/dyslexia/symptoms-causes/syc-20353552, 2022. (Accedido: 13-05-2023).

[3] Begoña Díaz Rincón. Definición, orígenes y evolución de la dislexia. [En línea]. Disponible en: https://summa.upsa.es/high.raw?id=0000029508&name=00000001.original.pdf, 2006. (Accedido: 13-05-2023).

[4] Luz Rello. Tipos de dislexia. [En línea]. Disponible en: https://blog.changedyslexia.org/dytective-luz-rello-tipos-de-dislexia/, 2020. (Accedido: 13-05-2023).

[5] Mª Cristina Pérez Pietri. 8 herramientas tecnológicas para niños con dislexia. [En línea] Disponible en: https://compartirenfamilia.com/tecnologia/8-herramientas-y-apps-para-ayudar-a-ninos-dislexicos.html, 2023. (Accedido: 18-05-2023).

[6] José Antonio Sánchez. ¿cómo aprenden las máquinas? machine learning y sus diferentes tipos. [En línea] Disponible en: https://datos.gob.es/es/blog/como-aprenden-las-maquinas-machine-learning-y-sus-diferentes-tipos, 2020. (Accedido: 18-05-2023).

[7] Instituto de Investigación de Ingeniería(IIC). Machine learning deep learning. los sistemas de ia aprenden de tus datos. [En línea] Disponible en: https://www.iic.uam.es/inteligencia-artificial/machine-learning-deep-learning/. (Accedido: 29-05-2023).

[8] Paloma Recuero de los Santos. Datos de entrenamiento vs datos de test: ¿en qué se diferencian? [En línea] Disponible en: https://empresas.blogthinkbig.com/datos-entrenamiento-vs-datos-de-test/, 2022. (Accedido: 18-05-2023).

[9] How machine learning is revolutionizing the healthcare industry. [En línea]. Disponible en: https://www.datasciencecentral.com/how-machine-learning-is-revolutionizing-the-healthcare-industry/, 2023. (Accedido: 18-05-2023).

[10] Universidad de Málaga (UMA). Utilizan técnicas de inteligencia artificial para el diagnóstico precoz de la dislexia. [En línea] Disponible en: https://www.uma.es/sala-de-prensa/noticias/

utilizan-tecnicas-de-inteligencia-artificial-para-el-diagnostico-precoz-de-la-dislexia/. (Accedido: 29-05-2023).

[11] William Villegas-Ch, Milton Román-Cañizares, and Xavier Palacios-Pacheco. Improvement of an online education model with the integration of machine learning and data analysis in an lms. *Applied Sciences*, 10(15), 2020.

[12] L. Rello, R. Baeza-Yates, A. Ali, J. P. Bigham, and M. Serra. Predicting risk of dyslexia with an online gamified test. *PLoS One*, 15(12):e0241687, 2020. PMID: 33264301; PMCID: PMC7710040.

[13] Wikipedia. Historia de python. [En línea] Disponible en: `https://es.wikipedia.org/wiki/Historia_de_Python`. (Accedido: 29-05-2023).

[14] Python Software Foundation. Python wiki: Beginners guide overview. [En línea] Disponible en: `https://wiki.python.org/moin/BeginnersGuide/Overview`. (Accedido: 18-05-2023).

[15] Andrés Visus. ¿para qué sirve python? razones para utilizar este lenguaje de programación. [En línea] Disponible en: `https://www.esic.edu/rethink/tecnologia/para-que-sirve-python`, 2020. (Accedido: 29-05-2023).

[16] NumPy Contributors. Numpy. [En línea] Disponible en: `https://numpy.org/doc/stable/user/whatisnumpy.html`, 2022. (Accedido: 18-05-2023).

[17] Pandas Development Team. Pandas. [En línea] Disponible en: `https://pandas.pydata.org/about/`, 2023. (Accedido: 18-05-2023).

[18] Scikit-learn.

[19] Imbalanced learn Contributors. Imbalanced-learn. [En línea] Disponible en: `https://imbalanced-learn.org/stable/introduction.html`, 2022. (Accedido: 18-05-2023).

[20] Mozilla Developer Network. Introducción a django. [En línea] Disponible en: `https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction`, 2023. (Accedido: 18-05-2023).

[21] Django Software Foundation. Django security. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/topics/security/`, 2023. (Accedido: 18-05-2023).

[22] Django Software Foundation. Django authentication. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/topics/auth/`, 2023. (Accedido: 18-05-2023).

[23] Django Software Foundation. Django password management. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/topics/auth/passwords/`, 2023. (Accedido: 18-05-2023).

[24] Django Software Foundation. Django models. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/topics/db/models/`, 2023. (Accedido: 18-05-2023).

[25] Django Software Foundation. Django views. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/topics/http/views/`, 2023. (Accedido: 18-05-2023).

[26] Django Software Foundation. Django applications. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/ref/applications/`, 2023. (Accedido: 18-05-2023).

[27] Django Software Foundation. Django urls. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/topics/http/urls/`, 2023. (Accedido: 18-05-2023).

[28] Django Software Foundation. Django templates. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/topics/templates/`, 2023. (Accedido: 18-05-2023).

[29] Django Software Foundation. Django sqlite notes. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/ref/databases/#sqlite-notes`, 2023. (Accedido: 19-05-2023).

[30] SQLite. Sqlite features. [En línea] Disponible en: `https://www.sqlite.org/features.html`, 2023. (Accedido: 19-05-2023).

[31] Django Software Foundation. Django admin documentation. [En línea] Disponible en: `https://docs.djangoproject.com/en/4.2/ref/contrib/admin/`, 2023. (Accedido: 19-05-2023).

[32] Torresburriel Estudio. Wcag 2.1: qué son y cómo respetarlas. [En línea] Disponible en: `https://www.torresburriel.com/weblog/2022/02/07/wcag-2-1-que-son-y-como-respetarlas/`, 2022. (Accedido: 29-05-2023).

[33] NVDA. Nvda. [En línea] Disponible en: `https://nvda.es/`, year = 2023, note = (Accedido: 19-05-2023),.

[34] TPG Interactive. Color contrast checker. [En línea] Disponible en: `https://www.tpgi.com/color-contrast-checker/`, 2023. (Accedido: 19-05-2023).

[35] VS Code Community. Microsoft. Visual studio code documentation. [En línea] Disponible en: `https://code.visualstudio.com/docs`, 2023. (Accedido: 19-05-2023).

[36] Giovanni Blandino. Figma: ¿qué es y para qué sirve? [En línea] Disponible en: `https://www.pixartprinting.es/blog/figma-que-es/`, 2023. (Accedido: 19-05-2023).

[37] Postman Contributors. What is postman? [En línea] Disponible en: `https://www.postman.com/product/what-is-postman/`, 2023. (Accedido: 19-05-2023).

[38] Wikipedia Contributors. Github. [En línea] Disponible en: `https://es.wikipedia.org/wiki/GitHub`, 2023. (Accedido: 19-05-2023).

[39] Diego de la Torre. Historia: Cómo nació la inteligencia artificial. [En línea] Disponible en: `https://blogthinkbig.com/historia-como-nacio-inteligencia-artificial/`, 2023. (Accedido: 19-05-2023).

[40] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943. (Accedido: 19-05-2023).

[41] Fernando Berzal. Inteligencia artificial. *Investigación y Ciencia*, 479:46–73, Agosto 2016. (Accedido: 19-05-2023).

[42] Victor Roman. Machine learning general process. [En línea] Disponible en: `https://towardsdatascience.com/machine-learning-general-process-8f1b510bd8af`, 2018. (Accedido: 20-05-2023).

[43] Graph Everywhere. Algoritmos de machine learning. [En línea] Disponible en: `https://www.grapheverywhere.com/algoritmos-de-machine-learning/`. (Accedido: 20-05-2023).

[44] Juan Francisco Vallalta Rueda. Aprendizaje supervisado y no supervisado. [En línea] Disponible en: `https://healthdataminer.com/data-mining/aprendizaje-supervisado-y-no-supervisado/`, 2023. (Accedido: 20-05-2023).

[45] Oscar Medina. Aprendizaje semi-supervisado. [En línea] Disponible en: https://www.predictiva.com.co/blog-predictiva/aprendizaje-semi-supervisado/, 2020. (Accedido: 20-05-2023).

[46] Na8. Aprendizaje por refuerzo. [En línea] Disponible en: https://www.aprendemachinelearning.com/aprendizaje-por-refuerzo/, 2020. (Accedido: 20-05-2023).

[47] Harikrishnan N B. Confusion matrix, accuracy, precision, recall, f1 score. [En línea] Disponible en: https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd, 2019. (Accedido: 20-05-2023).

[48] B. TRAVESÍ ZARAGOZA. Dislexia: un problema oculto que genera el fracaso escolar. [En línea] Disponible en: https://www.heraldo.es/noticias/aragon/2017/11/17/dislexia-problema-oculto-que-genera-del-fracaso-escolar-1208745-300.html, 2017. (Accedido: 19-05-2023).

[49] Claire Drumond. What is agile project management? [En línea] Disponible en: https://www.atlassian.com/agile/project-management, 2022. (Accedido: 29-05-2023).

[50] Max Rehkopf. What are sprints? [En línea] Disponible en: https://www.atlassian.com/agile/scrum/sprints, 2022. (Accedido: 29-05-2023).

[51] Kevin Ku. 7 ways to handle missing values in machine learning. [En línea] Disponible en: https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e, 2020. (Accedido: 29-05-2023).

[52] Chirag Goyal. Outlier detection removal — how to detect remove outliers. [En línea] Disponible en: https://www.analyticsvidhya.com/blog/2021/05/feature-engineering-how-to-detect-and-remove-outliers-with-python-code/, 2021. (Accedido: 29-05-2023).

[53] Harika Bonthu. Detecting and treating outliers — treating the odd one out! [En línea] Disponible en: https://www.analyticsvidhya.com/blog/2021/05/detecting-and-treating-outliers-treating-the-odd-one-out/, 2021. (Accedido: 29-05-2023).

[54] IBM. K-nearest neighbors algorithm (knn). [En línea] Disponible en: https://www.ibm.com/topics/knn. (Accessed on: 21-05-2023).

[55] IBM. What is logistic regression? [En línea] Disponible en: https://www.ibm.com/topics/logistic-regression. (Accedido: 21-05-2023).

[56] Anshul Saini. Support vector machine(svm): A complete guide for beginners. [En línea] Disponible en: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/, 2021.

[57] IBM. What is a decision tree? [En línea] Disponible en: https://www.ibm.com/topics/decision-trees. (Accedido: 21-05-2023).

[58] IBM. What is random forest? [En línea] Disponible en: https://www.ibm.com/topics/random-forest. (Accedido: 21-05-2023).

[59] Kurtis Pykes. Oversampling and undersampling. a technique for imbalanced classification. [En línea] Disponible en: `https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf`, 2020. (Accedido: 23-05-2023).

[60] Microsoft Azure Contributors. Smote. [En línea] Disponible en: `https://learn.microsoft.com/es-es/azure/machine-learning/component-reference/smote`, 2022. (Accedido: 23-05-2023).

[61] Organización de las Naciones Unidas. Transforming our world: The 2030 agenda for sustainable development. [En línea] Disponible en: `https://www.un.org/sustainabledevelopment/es/development-agenda/`, 2015. (Accedido: 26-05-2023).

[62] Organización de las Naciones Unidas. Objetivos de desarrollo sostenible. [En línea] Disponible en: `https://www.un.org/es/ga/president/65/issues/sustdev.shtml`. (Accedido: 26-05-2023).

[63] ONU. Objetivo 4: Educación de calidad. [En línea] Disponible en: `https://www.un.org/sustainabledevelopment/es/education/`, 2015. (Accedido: 31-05-2023).

[64] ONU. Objetivo 10: Reducir la desigualdades. [En línea] Disponible en: `https://www.un.org/sustainabledevelopment/es/inequality/`, 2015. (Accedido: 31-05-2023).

Anexos

Anexo A

Alineación del proyecto con los ODS

En el año 2015, los líderes mundiales adoptaron un acuerdo internacional sin precedentes: **la Agenda 2030**. Todos los Estados Miembros de las Naciones Unidas aprobaron diecisiete objetivos para alcanzar el desarrollo sostenible, los Objetivos de Desarrollo Sostenible (ODS) [61]. La lucha contra la pobreza, el cuidado del planeta y la disminución de las desigualdades son los principales propósitos que recoge la Agenda 2030. Cada uno de estos objetivos consta de una serie de metas que deben alcanzarse en los próximos diez años.

Figura A.1: Objetivos de Desarrollo sostenible de la Agenda 2030

Los Objetivos de Desarrollo Sostenible buscan alcanzar la sostenibilidad, definida como "la satisfacción de las necesidades de la generación presente sin comprometer la capacidad de las generaciones futuras

para satisfacer sus propias necesidades" [62].

Mediante esta página web, se pretende superar la barrera que existe entre los niños con dislexia y los niños sin dislexia. Los centros escolares no suelen estar adaptados para enseñar a niños con trastornos del aprendizaje, lo que resulta en un aumento del fracaso escolar.

El modelo de Machine Learning y el test diseñados permiten identificar si una persona tiene dislexia, brindándoles la oportunidad de abordarla tempranamente y aumentando la posibilidad de superarla a tiempo. Esto asegura la igualdad de oportunidades para aquellos que tienen dislexia.

Además, los ejercicios diseñados para trabajar las competencias afectadas por la dislexia tienen en cuenta las necesidades de las personas cuyo cerebro no trabaja a la velocidad esperada. Se desarrollarán en un entorno interactivo y cercano, donde los usuarios puedan disfrutar mientras trabajan, lo que facilita la superación de la dislexia.

Analizando la descripción de cada uno de los diecisiete objetivos, se ha llegado a la conclusión de que los dos que presentan mayor relación con el propósito del desarrollo de este proyecto son:

ODS4. Educación de calidad

El Objetivo de Desarrollo Sostenible número 4 se enfoca en garantizar una educación inclusiva, equitativa y de calidad para todos. Esta meta busca construir y adecuar instalaciones educativas que sean inclusivas y seguras para todos los niños, incluyendo aquellos con discapacidades y diferencias de género [63].

En el contexto de este proyecto, se busca contribuir al logro del ODS4 al proporcionar una herramienta que aborda la dislexia, un trastorno del aprendizaje que puede dificultar la educación de calidad para aquellos que lo padecen. Al identificar la dislexia de manera temprana y brindar ejercicios y recursos diseñados específicamente para abordar las dificultades asociadas con este trastorno, se espera mejorar las oportunidades de aprendizaje y promover una educación inclusiva y equitativa.

Además, al ofrecer un entorno interactivo y cercano en la aplicación web, se busca crear un ambiente seguro y propicio para el aprendizaje de los usuarios. Esto se alinea con la meta 4.a, que busca garantizar instalaciones educativas que sean inclusivas y seguras para todos los niños.

ODS10. Reducción de desigualdades

El Objetivo de Desarrollo Sostenible número 10 se centra en reducir las desigualdades dentro y entre los países. La meta 10.3 busca garantizar la igualdad de oportunidades y reducir la desigualdad de resultados, incluyendo la eliminación de leyes, políticas y prácticas discriminatorias [64].

Este proyecto contribuye a la meta 10.3 al abordar las desigualdades en el ámbito educativo relacionadas con la dislexia. La detección temprana y el tratamiento adecuado de la dislexia pueden ayudar a reducir la desigualdad de resultados en el aprendizaje y promover la igualdad de oportunidades para las personas con este trastorno del aprendizaje.

Además, al proporcionar herramientas y recursos específicos para abordar la dislexia, este proyecto busca eliminar las barreras y discriminaciones que pueden existir en el sistema educativo. Al enfocarse en las necesidades de las personas con dislexia y promover su inclusión y participación en la educación, se busca contribuir a la reducción de las desigualdades en el acceso y los resultados educativos.

Adicionalmente, al diseñar la aplicación web con consideraciones de accesibilidad en mente, se busca asegurar que todas las personas, incluyendo aquellas con discapacidades o necesidades especiales, puedan acceder y utilizar la herramienta de manera efectiva. Esto implica proporcionar opciones de navegación alternativas, compatibilidad con tecnologías de asistencia, ajustes de contraste y tamaño de texto, y otras características que mejoren la usabilidad para diferentes usuarios.

La accesibilidad se convierte en un factor clave para lograr una educación inclusiva y de calidad. Al garantizar que la herramienta sea accesible para todos los usuarios, independientemente de sus capacidades o limitaciones, se promueve la igualdad de oportunidades y se fomenta la participación activa de todas las personas en el proceso educativo.

En resumen, este proyecto se alinea con los ODS 4 y 10 al abordar la educación inclusiva y de calidad, así como la reducción de desigualdades, especialmente en el ámbito de la dislexia. Al asegurar que la aplicación web sea accesible para todos los usuarios, se contribuye a una educación inclusiva y se garantiza que todas las personas, incluyendo aquellas con discapacidades o dificultades de aprendizaje, puedan beneficiarse de la herramienta y participar plenamente en el proceso educativo.

Anexo B

Manual de instalación

El Manual de Instalación detalla, paso a paso, las pautas necesarias para instalar y configurar el entorno de trabajo de la aplicación. En el Capítulo 2 se especificaron todas las tecnologías que se han utilizado para desarrollar exitosamente la aplicación. A continuación, se muestra cómo instalar los *software* más relevantes.

B.1. Python

El *software* de Python se puede descargar directamente desde la página web ofocial del mismo: `https://www.python.org/`. Esta presenta un menú de navegación, tal y como se muestra en la Figura B.1 donde se ofrece la opción de *Downloads*. En función de la versión que se desee o el sistema operativo en el que se esté trabajando, pudiendo elegir entre distintas opciones.

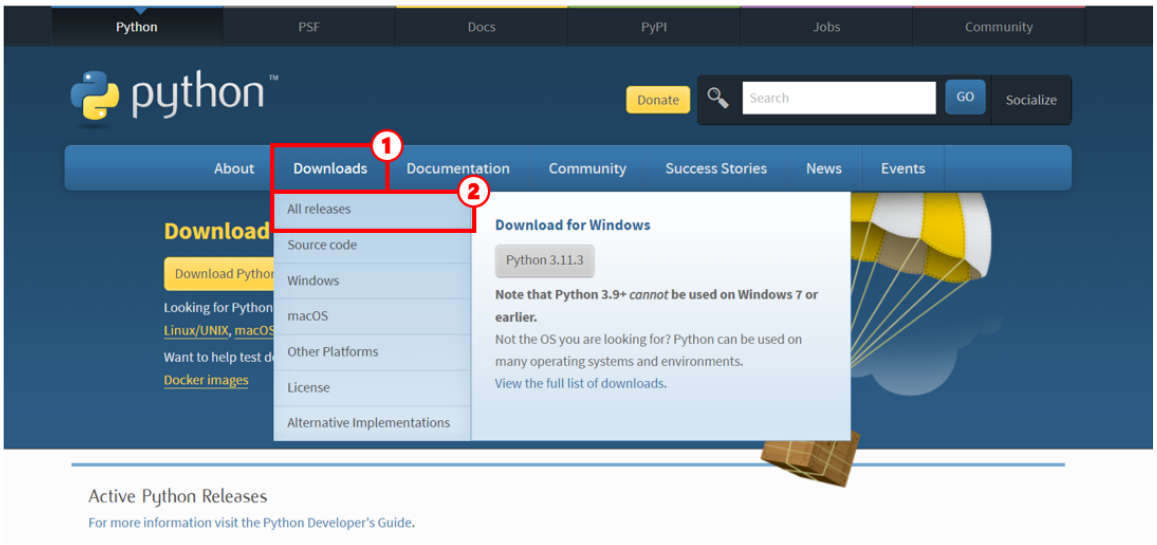


Figura B.1: Instalación de Pyhton - Paso 1

En este caso se va a acceder a *All releases* con el objetivo de elegir la versión *Python 3.8.13* como se

muestra en la Figura B.2, ya que es la versión de Python en la que se ha desarrollado la aplicación. No obstante, se podría elegir una más reciente si así se desea.

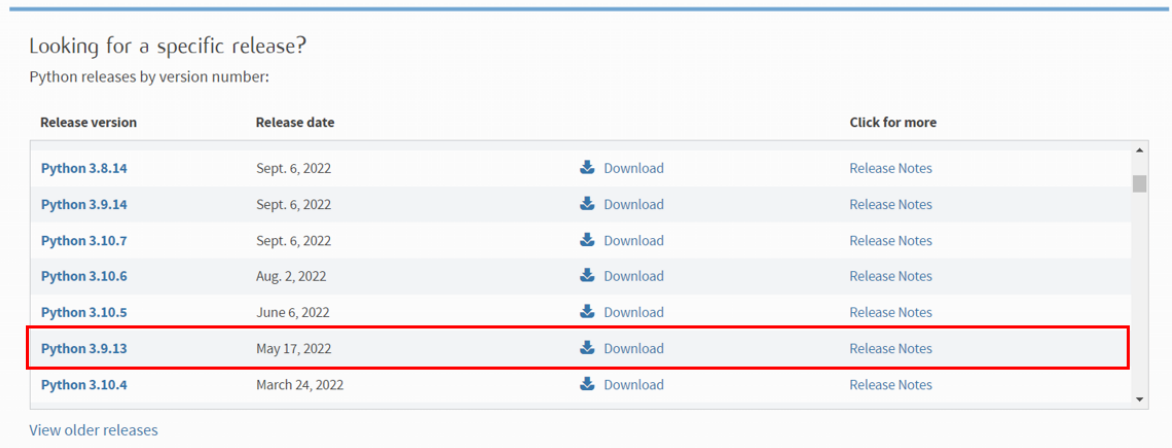


Figura B.2: Instalación de Pyhton - Paso 2

Tras pulsar *Download* la página oficial de Python redirige a otra vista donde se podrá elegir que versión se quiere descargar. En el caso de la Figura B.3 se ha seleccionado la última opción ya que se está trabajando en Windows.

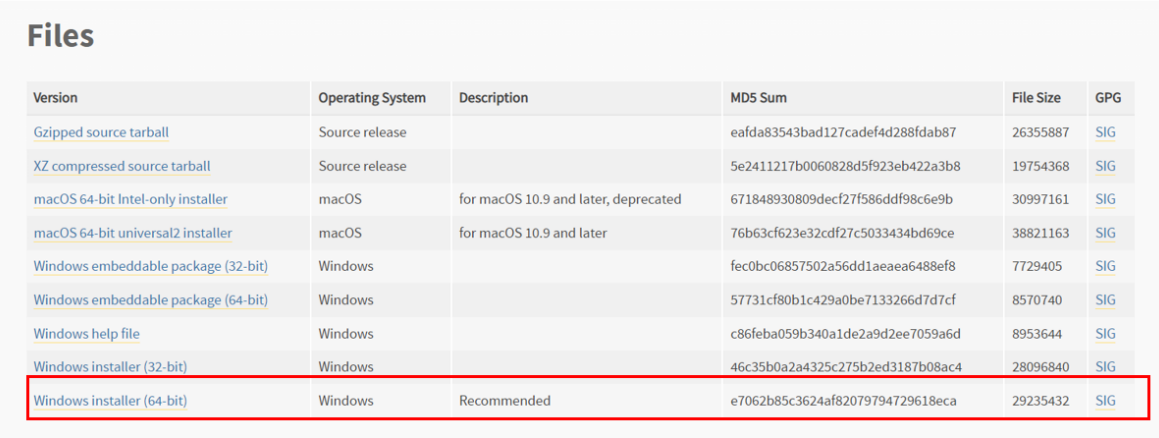


Figura B.3: Instalación de Pyhton - Paso 3

Se descargará un archivo con la extensión *.exe*, el cual se tendrá que ejecutar para poder instalar correctamente Python. Se mostrará por pantalla un panel con dos opciones, de las cuales se recomienda elegir *Install now*, verificando que se han marcado las casillas *Install launcher for all users* y *Add Python 3.9 to PATH*. Tras unos segundos se completará la instalación y se podrá comprobar que no ha ocurrido ningún problema buscando en las aplicaciones del sistema:

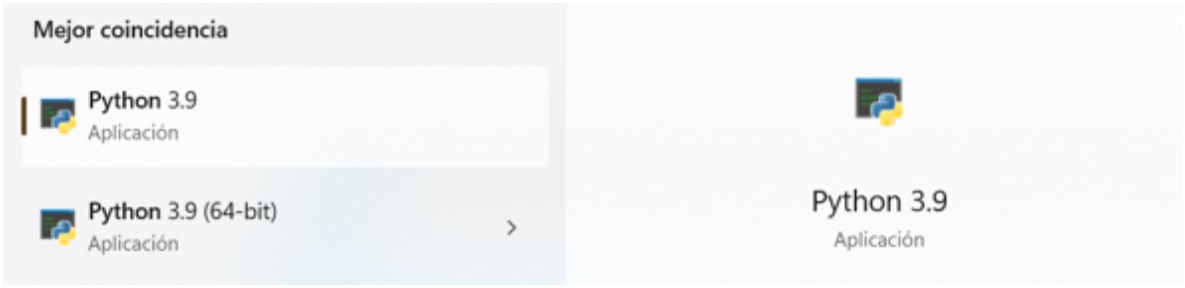


Figura B.4: Instalación de Pyhton - Paso 4

A su vez, se debe comprobar que la versión instalada es la elegida y que funciona correctamente. Esto se puede hacer abriendo la terminal e introduciendo los comandos `Python --version` y `Pyhton`:

```
PS C:\Users\blanc> python --version
Python 3.9.13
PS C:\Users\blanc> python
Python 3.9.13 (tags/v3.9.13:6de2ca5, May 17 2022, 16:36:42) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello world")
Hello world
>>> exit()
PS C:\Users\blanc>
```

Figura B.5: Instalación de Pyhton - Paso 5

Finalmente, solo quedaría instalar las librerías empleadas para manejar los datos y diseñar el modelo de Machine Learning. Para instalar librerías de Python, se puede emplear `pip`. Este comando permite la gestión e instalación de paquetes de Python. Para instalarlo es necesario seguir una serie de pasos:

- Descargar el `script get-pip.py` desde la dirección `https://bootstrap.pypa.io/get-pip.py`
- Abrir una terminal o línea de comandos y acceder al directorio donde se ubica el archivo *get-pip.py*.
- Ejecutar el comando `python get-pip.py`

Para comprobar si se ha instalado correctamente se comprueba la versión mediante `pip --version`:

```
PS C:\Users\blanc> pip --version
pip 23.1.2 from C:\Users\blanc\env\lib\site-packages\pip (python 3.9)
```

Figura B.6: Instalación de Pyhton - Paso 6

Cabe la posibilidad de que sean necesarios permisos de administrador para instalar `pip`. En ese caso, en sistemas Unix y MacOS, se debe utilizar `sudo python get-pip.py`, mientras que en sistemas Windows será necesario abrir la terminal como administrador.

Las librerías se instalarán con el comando `pip install nombre_de_la_librería`. Este comando sirve para instalar cualquier librería de Python.

B.2. Django

Previo a la instalación de Django, es necesario verificar que Python está instalado correctamente en el dispositivo. A su vez, se recomienda tener instalado un editor de texto. El proyecto se ha desarrollado empleando Visual Studio Code y más adelante se detallará como tenerlo funcionando en el dispositivo.

Se va a crear un entorno virtual, lo que permite instalar y gestionar paquetes en el proyecto que se está trabajando de forma aislada. Para ello se va a ejecutar el comando `pyhton -m venv env`, como se muestra en la Figura B.7. A continuación, se va a iniciar el entorno virtual mediante `env\Scripts\active`

```
PS C:\Users\blanc> cd C:\Users\blanc\ICAI_4\TFG\my_django_app
PS C:\Users\blanc\ICAI_4\TFG\my_django_app> python -m venv env
PS C:\Users\blanc\ICAI_4\TFG\my_django_app> env\Scripts\activate
(env) PS C:\Users\blanc\ICAI_4\TFG\my_django_app>
```

Figura B.7: Instalación de Django - Paso 1

Si se accede al directorio donde se está trabajando se observa como se ha creado un directorio llamado `env`, el cual incluye una carpeta llamada `Scripts`:

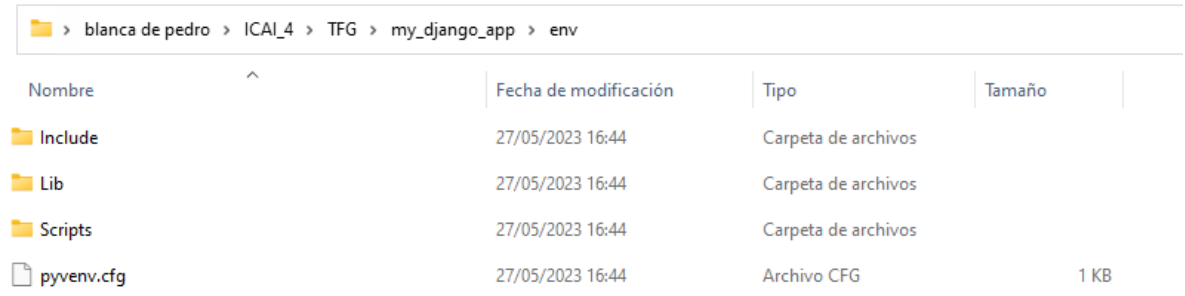


Figura B.8: Instalación de Django - Paso 2

Una vez se tiene el entorno virtual funcionando correctamente se va a proceder con la instalación de Django a través del comando `pip install django`:

```
(env) PS C:\Users\blanc\ICAI_4\TFG\my_django_app> pip install django
Collecting django
  Downloading Django-4.2.1-py3-none-any.whl (8.0 MB)
     ---------------------------------------- 8.0/8.0 MB 26.9 MB/s eta 0:00:00
Collecting tzdata
  Using cached tzdata-2023.3-py2.py3-none-any.whl (341 kB)
Collecting asgiref<4,>=3.6.0
  Downloading asgiref-3.7.1-py3-none-any.whl (24 kB)
Collecting sqlparse>=0.3.1
  Downloading sqlparse-0.4.4-py3-none-any.whl (41 kB)
     ---------------------------------------- 41.2/41.2 kB 2.1 MB/s eta 0:00:00
Collecting typing-extensions>=4
  Downloading typing_extensions-4.6.2-py3-none-any.whl (31 kB)
Installing collected packages: tzdata, typing-extensions, sqlparse, asgiref, django
Successfully installed asgiref-3.7.1 django-4.2.1 sqlparse-0.4.4 typing-extensions-4.6.2 tzdata-2023.3
```

Figura B.9: Instalación de Django - Paso 3

Para iniciar un proyecto nuevo en Django se va a ejecutar en la terminar el comando `django-admin stratproject my_django_app`, donde *my_django_app* es el nombre de la aplicación:

```
(env) PS C:\Users\blanc\ICAI_4\TFG\my_django_app> django-admin startproject my_django_app
(env) PS C:\Users\blanc\ICAI_4\TFG\my_django_app> cd my_django_app
(env) PS C:\Users\blanc\ICAI_4\TFG\my_django_app\my_django_app>
```

Figura B.10: Instalación de Django - Paso 4

Finalmente, para acceder a la aplicación que se acaba de crear se ejecuta en la terminal `python manage.py runserver`. El comando se tiene que ejecutar dentro de la aplicación web, si no no encontrará el fichero *manage.py* y no se podrá iniciar la aplicación.

```
(env) PS C:\Users\blanc\ICAI_4\TFG\my_django_app\my_django_app> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.

Run 'python manage.py migrate' to apply them.
May 27, 2023 - 16:53:05
Django version 4.2.1, using settings 'my_django_app.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[27/May/2023 16:53:07] "GET / HTTP/1.1" 200 10731
[27/May/2023 16:53:07] "GET /static/admin/css/fonts.css HTTP/1.1" 404 1816
```

Figura B.11: Instalación de Django - Paso 5

Al ejecutar el comando, se proporciona por la terminal la dirección del servidor donde se ha desplegado la aplicación. Si se accede a dicha dirección se obtiene:

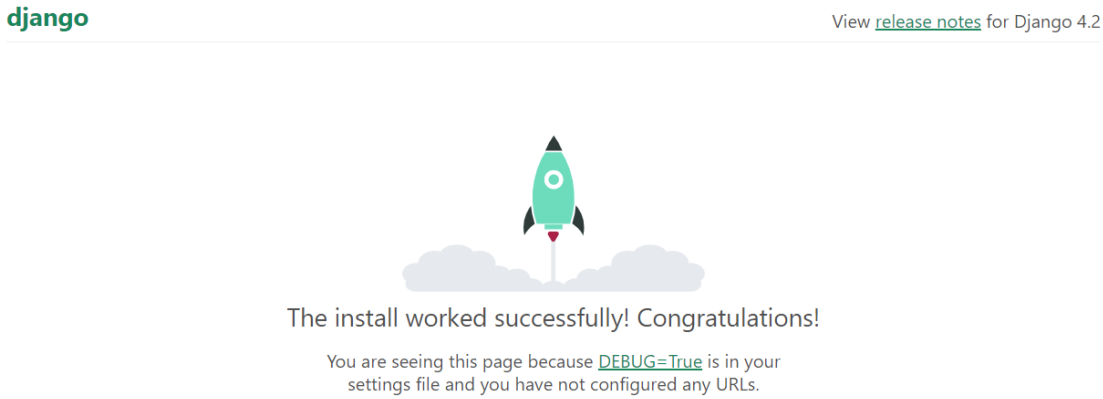


Figura B.12: Instalación de Django - Paso 6

Esto indica que ya se tiene Django funcionando y se puede empezar con el desarrollo de la aplicación.

B.3. NVDA

NVDA (*NonVisual Desktop Access*) es un lector de pantalla gratuito y de código abierto para el sistema operativo Windows. Accediendo a su página oficial `https://nvda.es/descargas/descarga-de-nvda/`. Tras descargar el fichero ejecutable aparecerá por pantalla:

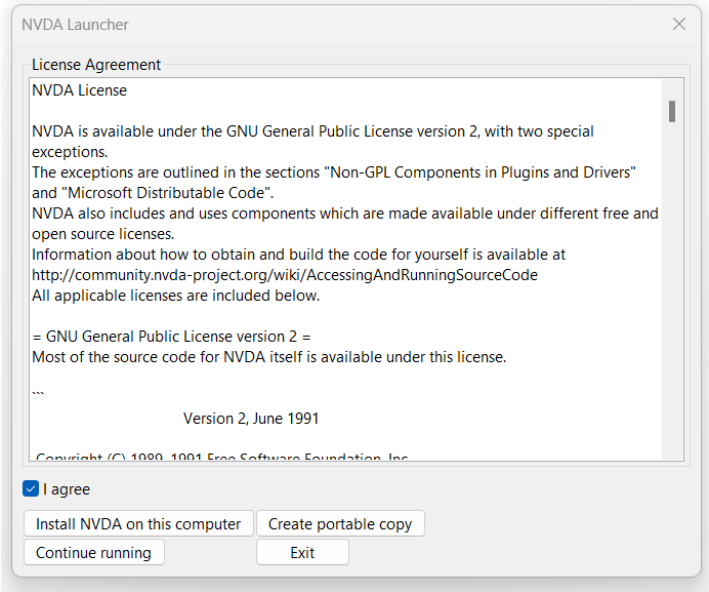


Figura B.13: Instalación de NVDA - Paso 1

Es necesario verificar que se está dando el consentimiento a la licencia. Una vez se haya completado la instalación estará preparado para su uso. NVDA ofrece un gran variedad de opciones para ajustar la disposición de la herramienta de forma personal, las cuales se pueden modificar desde el menú principal:

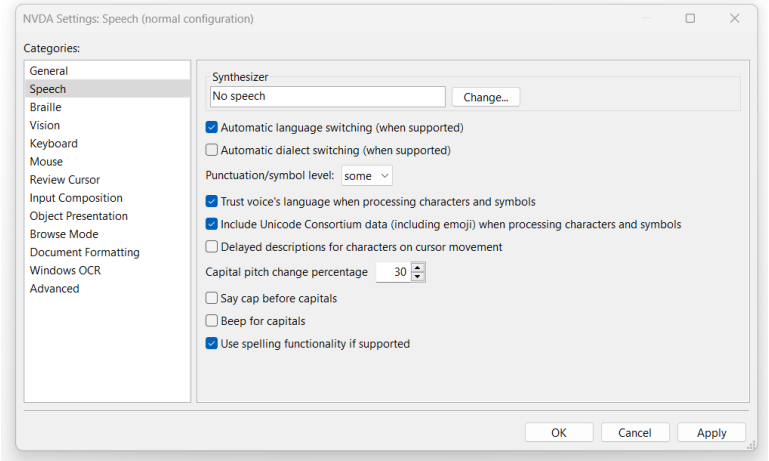


Figura B.14: Instalación de NVDA - Paso 2

B.4. Visual Studio Code

Es necesario utilizar un editor de texto para el desarrollo del proyecto y en este caso se ha elegido Visual Studio Code. Para su instalación habrá que acceder a la página oficial en `https://code.visualstudio.com/` e instalar la versión que se adapte al sistema operativo que se está usando.

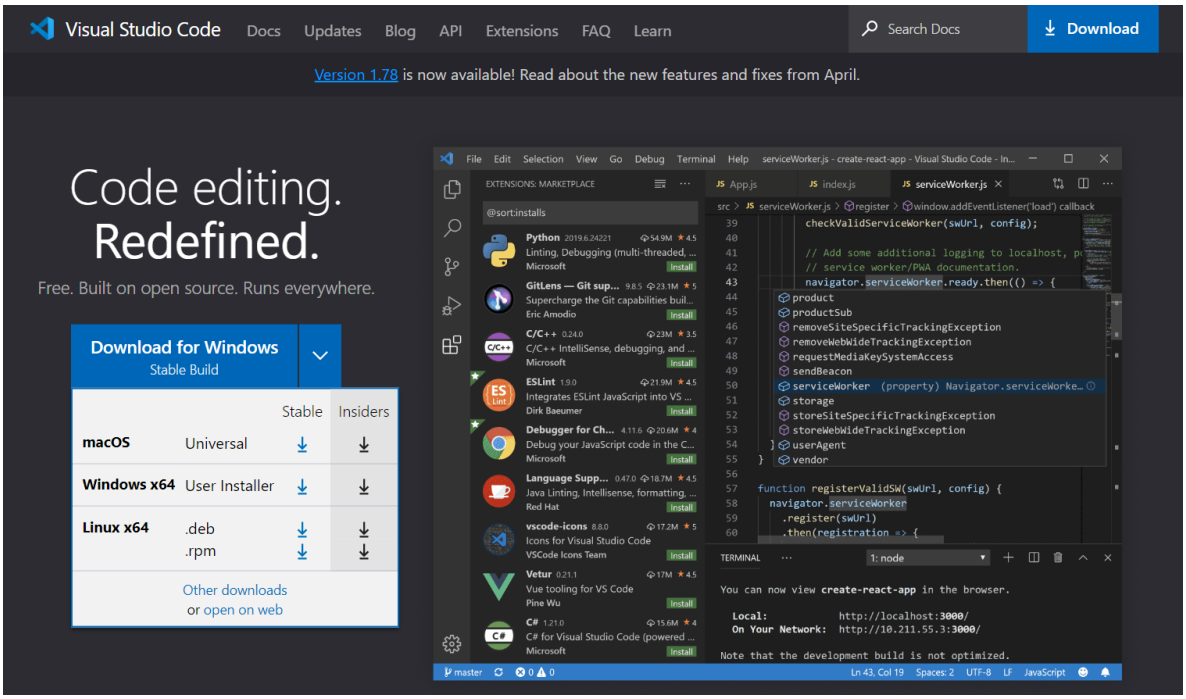


Figura B.15: Instalación de Visual Studio Code

A continuación, se ejecutará el instalador y habrá que seguir las instrucciones que aparecen por pantalla. Es importante verificar que se ha marcado la casilla "*Add to PATH*" durante la instalación. Una vez finalizada la instalación Visual Studio Code estará listo para uso.

B.5. Instalación del proyecto

Se va a proceder con la descarga del proyecto desarrollado, para lo que se va a hacer uso de GitHub. El Trabajo Fin de Grado se ha alojado en GitHub, plataforma que ha permitido llevar un seguimiento de las versiones del mismo.

Si se es familiar con el concepto de Git y se tiene instalado en el dispositivo se procederá con la clonación del repositorio de GitHub en el ordenador mediante el comando `git clone https://github.com/BlancadePedro/TFG` en el directorio en el que se quiera trabajar. De lo contrario, se puede acceder a `https://github.com/BlancadePedro/TFG` y descargarse un archivo comprimido. El contenido que se obtiene es el mismo de ambas formas.

```
PS C:\Users\blanc> git clone https://github.com/BlancadePedro/TFG
Cloning into 'TFG'...
remote: Enumerating objects: 10615, done.
remote: Counting objects: 100% (526/526), done.
remote: Compressing objects: 100% (292/292), done.
remote: Total 10615 (delta 244), reused 486 (delta 211), pack-reused 10089 eceiving objects: 100% (10615/10615), 120.84
Receiving objects: 100% (10615/10615), 128.21 MiB | 17.08 MiB/s, done.

Resolving deltas: 100% (4983/4983), done.
Updating files: 100% (12662/12662), done.
```

Figura B.16: Clonación de repositorio en GitHub

Si se accede a `C:\Users\blanc\TFG> ls` se comprueba que contiene todos los repositorios del proyecto:

```
PS C:\Users\blanc\TFG> ls


    Directorio: C:\Users\blanc\TFG


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        27/05/2023     18:53                data_and_ML_model
d-----        27/05/2023     18:53                dyslexia
d-----        27/05/2023     18:53                practica
d-----        27/05/2023     18:53                prueba
d-----        27/05/2023     18:53                register
d-----        27/05/2023     18:53                static
d-----        27/05/2023     18:53                templates
d-----        27/05/2023     18:53                TFG
-a----        27/05/2023     18:53         167936 db.sqlite3
-a----        27/05/2023     18:53            681 manage.py
-a----        27/05/2023     18:53         978946 ml_model.sav
-a----        27/05/2023     18:53           1606 package-lock.json
-a----        27/05/2023     18:53            318 package.json
-a----        27/05/2023     18:53             95 pytest.ini
```

Figura B.17: Contenido del proyecto obtenid através de GitHub

Antes de desplegar y acceder al contenido de la aplicación es necesario instalar una serie de paquetes que se han empleado para proporcionar una mejor experiencia de usuario. Se va a emplear el comando `pip`:

- Crispy Forms: `pip install django-crispy-forms`
- Crispy Bootstrap4: `pip install bootstrap4`
- Fontawesome Free: `pip install fontawesomefree`

Finalmente, se ejecutará el comando `python manage.py runserver`que proporcionará la dirección del servidor `http://127.0.0.1:8000/` mediante el cual se puede acceder a la aplicación web.

```
PS C:\Users\blanc\ICAI_4\TFG\TFG> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 27, 2023 - 19:09:44
Django version 4.1.1, using settings 'TFG.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Figura B.18: Resultado de la ejecución de *python manage.py runserver*

Anexo C

Manual de usuario

Se va a exponer una guía detallada sobre cómo utilizar la aplicación web, con explicaciones de cada sección. Cuando se accede por primera vez se muestra la página de inicio, la cual tiene información sobre las funcionalidades que se ofrece y sobre el proceso de creación del proyecto.

Figura C.1: Página de inicio

A partir de esta vista se tiene acceso a todo el contenido que ofrece la aplicación. No solo por los enlaces que se muestran en la cabecera y el pie de página, los cuales están presentes en todo momento, a excepción de cuando se realiza una actividad concreta, sino que se muestra un carrusel que permite acceder a las distintas páginas que incluye la aplicación, incluyendo una pequeña introducción de las mismas:

Figura C.2: Información sobre la página web

La aplicación web permite una navegación fluida e intuitiva en todo momento. Se puede acceder a cada sección con total libertad.

- **Productos**: en esta vista se exponen detalladamente las distintas funcionalidades de la aplicación. Esta sección es puramente informativa y el objetivo es que el usuario sea consciente en todo momento de qué producto se le ofrece y por qué destaca dicho producto.
- **Test**: a través de esta vista se accede a la prueba de detección de dislexia. Se ofrece al usuario contexto sobre el modelo de Machine Learning y al final de la página se muestra un enlace, claramente diferenciado, que permite el acceso al test. Cabe destacar que la prueba solo se puede realizar una vez. Por ello, si ya se ha realizado y se pretende acceder al enlace mencionado, se mostrarán por pantalla los resultados obtenidos, impidiendo que un usuario pueda realizar el test más de una vez. Los ejercicios que conforman el test se describen en el Anexo D.
- **Práctica**: a través de esta vista se accede a los ejercicios de entrenamiento que se proponen para el trabajo continuo de la dislexia, buscando la superación de la misma. Se presentan seis imágenes, cada una asociada a un ejercicio. Existe total libertad en la navegación de los ejercicios, pudiendo realizar uno más de una vez de forma continuada o intermitente. Los ejercicios que se presentan para el tratamiento de la dislexia se explican detalladamente en el Anexo E

La aplicación ofrece dos funcionalidades adicionales asociadas directamente con la experiencia de usuario. Los usuarios pueden acceder a su información personal, inicialmente introducida al registrarse en la aplicación, la cual podrán mantener actualizada en todo momento.

Cada vez que los usuarios inicien sesión o se registren por primera vez podrán acceder a su perfil y modificar la información que se les presenta. Cada campo se muestra de forma individual, claramente diferenciado por una etiqueta, junto a un botón (icono lapicero). Si se desea modificar la información que se presenta en el perfil, basta con reescribirla y pulsar dicho botón para que se actualice en la base de datos.

Figura C.3: Perfil del Usuario

De la misma forma, se muestra la opción de eliminar la cuenta, que requiere de una confirmación:

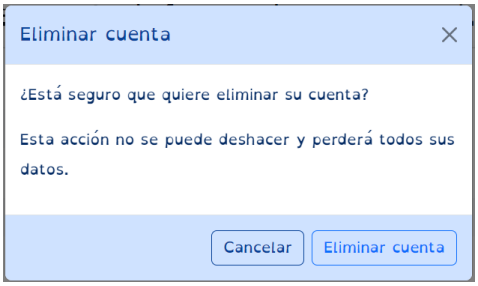


Figura C.4: Confirmación antes de eliminar una cuenta de usuario

Y resetear la contraseña, donde habrá que introducir la contraseña actual y por la que se quiere actualizar:

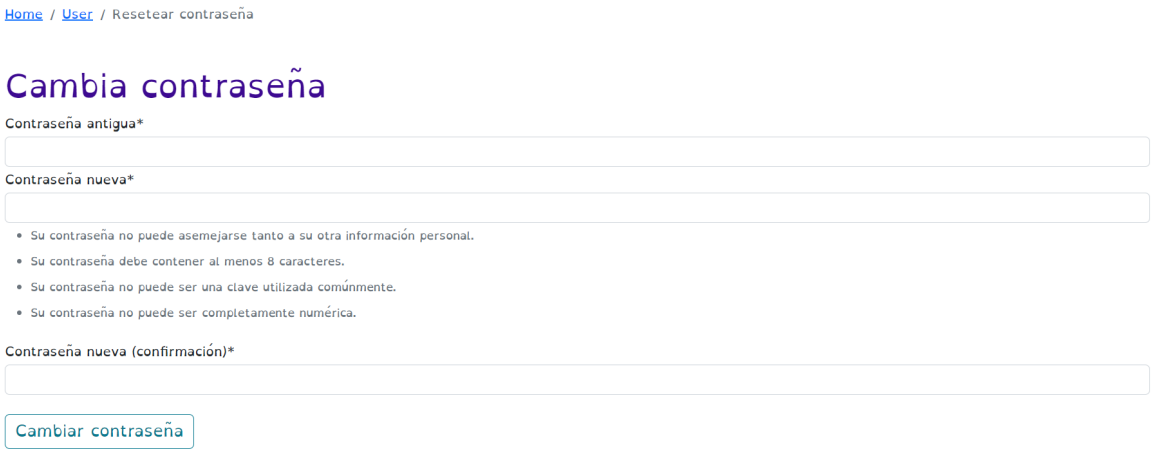


Figura C.5: Actualización de la contraseña

Para acceder a las diferentes funcionalidades de la aplicación el usuario debe estar autenticado, lo que implica que haya iniciado sesión o que se registre por primera vez. Los campos que debe rellenar vienen acompañados por aclaraciones y se muestra de forma clara y visual que información es obligatoria y aún no se ha completado:

Figura C.6: Registro de un nuevo usuario

La aplicación permite al usuario recuperar su cuenta en caso de que haya olvidado la contraseña. Es importante que el usuario tenga un correo electrónico preestablecido, ya que el sistema buscará en la base de datos dicho correo y modificará la cuenta de usuario que coincida. Si no se tiene un correo asociado a la cuenta, no se podrá recuperar la contraseña de forma segura.

Figura C.7: Paso 1 para recuperar la cuenta

Tras dar a enviar se mostrará por pantalla un mensaje informando de que se ha enviado un correo al correo proporcionado.

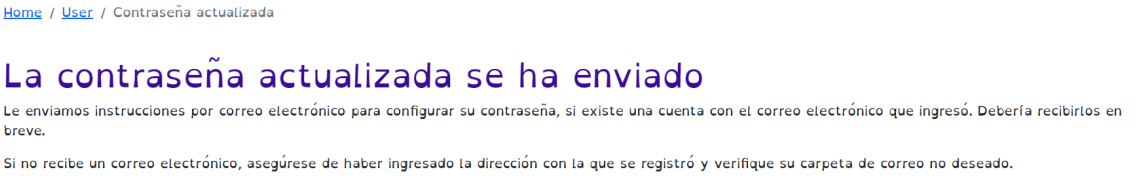


Figura C.8: Paso 2 para recuperar la cuenta

Una vez se reciba dicho correo, será necesario seguir el enlace proporcionado. En enlace redirige al

usuario a una nueva vista de la aplicación web que le permitirá introducir una contraseña nueva, siempre respetando las validaciones de autenticación ya mencionadas en las secciones anteriores.

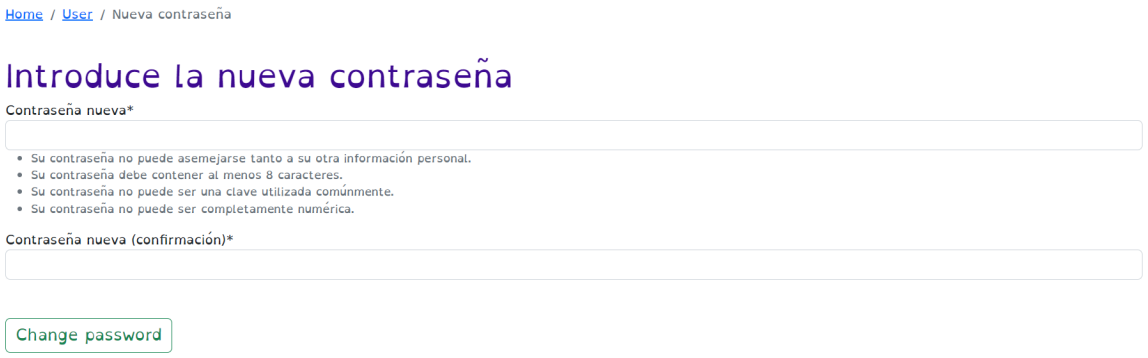


Figura C.9: Paso 3 para recuperar la cuenta

Finalmente, se mostrará un mensaje de éxito, el cual permitirá acceder al inicio de la aplicación e iniciar sesión con la nueva configuración.

Figura C.10: Paso 4 para recuperar la cuenta

Por otro lado, la aplicación ofrece un menú que permite al usuario personalizar la interfaz de la aplicación, permitiendo adaptarla a sus necesidades, tal y como se expone en el capítulo 6. El menú se presenta en todo momento mediante la etiqueta *Ajustes*:

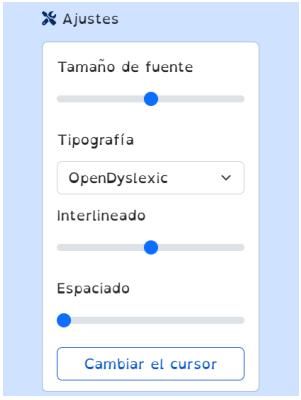


Figura C.11: Menú de ajustes

Anexo D

Test de detección de dislexia

La principal funcionalidad que ofrece la página web es el test de detección de dislexia, mediante el cual se determina si una persona tiene o no dislexia. La prueba consta de 32 cuestiones diferentes, evaluando en cada una de ellas un conjunto de procesos cognitivos y almacenando los resultados necesarios para poder determinar si el usuario padece o no dislexia.

La lógica de los ejercicios se ha implementado mediante JavaScript, empleando un archivo JSON que contiene la información necesaria para construir la base del ejercicio. Diferenciadas por cada índice, se han almacenado las distintas combinaciones de letras, sílabas, palabras y audios necesarias para construir el test completo. El único elemento que todos los ejercicios tienen en común, aparte del índice, es el campo "*url_audio*", que contiene la dirección para acceder al enunciado auditivo del ejercicio. Este enunciado solo se podrá escuchar una vez en la vista previa antes de la resolución del ejercicio.

Cuando el usuario acceda al test, se le mostrarán por pantalla una serie de instrucciones que deberá tener en cuenta para la correcta realización de los ejercicios. Se le proporciona una prueba para comprobar que el audio de su dispositivo se escucha adecuadamente. Una vez completada esta verificación, el usuario deberá pulsar el botón " Iniciar test " para comenzar el test.

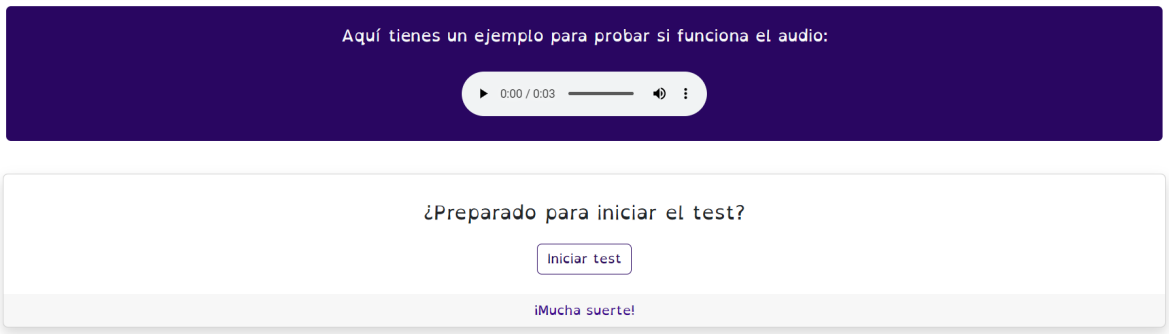


Figura D.1: Instrucciones del test

D.1. Ejercicios del 1 al 13 y del 18 al 21

En los primeros ejercicios del test, se presenta una cuadrícula cuyas dimensiones varían para aumentar la dificultad de manera progresiva. Esta cuadrícula está compuesta por un conjunto de letras, sílabas o palabras, reales o inventadas, dispuestas de tal manera que cada elemento funcione como un botón que reaccione cuando el cursor pase por encima. El usuario deberá pulsar el elemento mencionado en el enunciado tantas veces como le sea posible. Independientemente de si acierta o no, cada vez que se presione un elemento, los demás se redistribuirán de forma aleatoria, obligando al usuario a buscar continuamente la letra, sílaba o palabra mencionada. Se garantiza que el elemento mencionado esté presente en todo momento, aunque puede aparecer más de una vez debido a la disposición aleatoria.

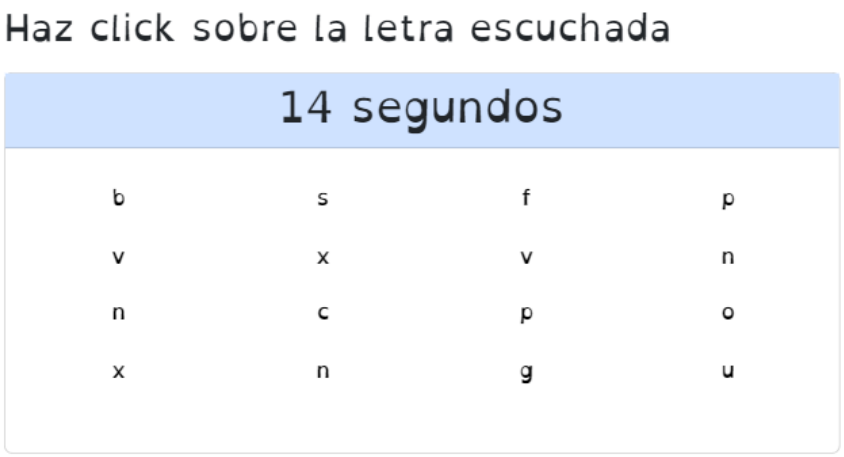
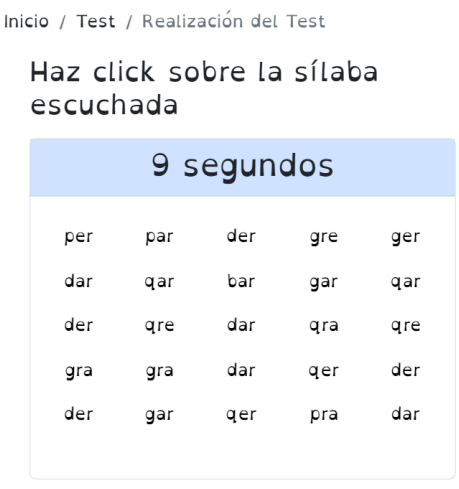
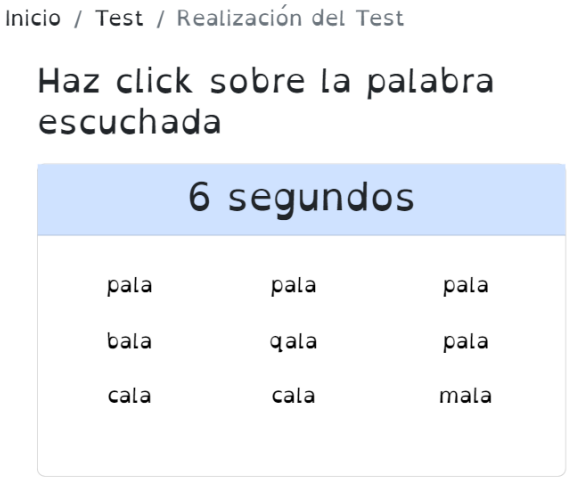


Figura D.2: Ejercicio 2



(a) Ejercicio 6



(b) Ejercicio 10

Figura D.3: Resolución 665 x 1045

D.2. Ejercicios del 14 al 17

En este conjunto de ejercicios, se presenta una cuadrícula de 5x5 compuesta por la misma letra repetida, a excepción de una, que el usuario deberá encontrar. El objetivo es localizar la letra diferente tantas veces como sea posible. Al igual que en los ejercicios anteriores, cada vez que se presione un elemento, la cuadrícula se redistribuirá de forma aleatoria, lo que obligará al usuario a buscar continuamente la letra distinta.

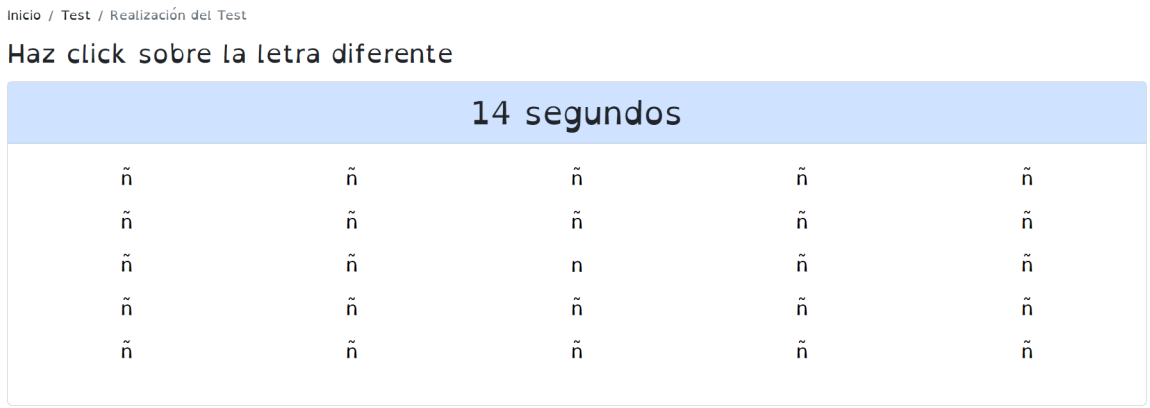


Figura D.4: Ejercicio 17

Cada uno de los cuatro ejercicios tiene un elemento *clave:valor* asociado, donde la clave es la letra distinta que el usuario debe encontrar y el valor la letra que se va a repetir. Las letras pueden ser sílabas, consonantes, una mezcla de ambas, mayúsculas o minúsculas.

D.3. Ejercicio 22

El ejercicio 22 consta de dos elementos: la palabra incompleta que el usuario debe completar y las opciones proporcionadas para ello. Las opciones de letras se presentan en una fila de botones, de los cuales solo uno de los cinco elementos es correcto. El usuario deberá completar correctamente tantas palabras como le sea posible. Cada vez que se pulse un botón, ya sea que el usuario haya acertado o no, se mostrará la siguiente palabra incompleta almacenada en el archivo JSON.

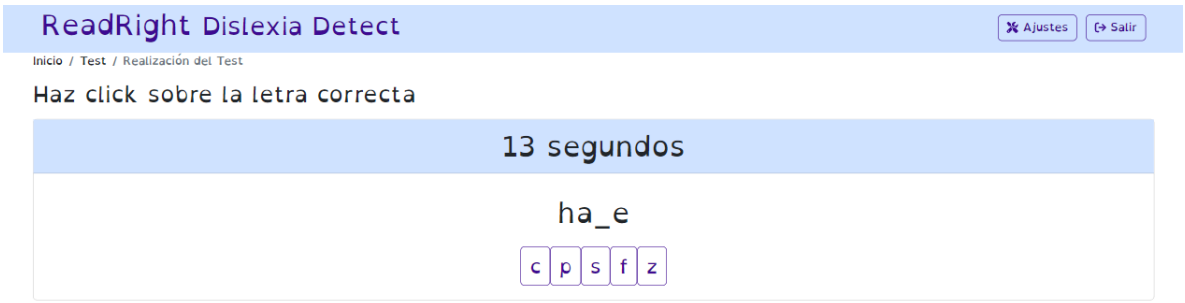


Figura D.5: Ejercicio 22

A su vez, en el archivo JSON, se almacena la letra correcta en el argumento "*missing*". Para comprobar si el usuario ha acertado, se comparará el valor del elemento pulsado con el valor de "*missing*", sumando puntos a los aciertos si coinciden o a los fallos si no lo hacen.

D.4. Ejercicio 23

En el ejercicio 23, se presenta una palabra escrita incorrectamente debido a que contiene una letra de más. Cada letra es un elemento clickeable que reacciona cuando el cursor pasa por encima. El objetivo es que el usuario identifique cuál es la letra extra. Cada vez que se pulse una letra, independientemente de si ha encontrado la que sobra o no, se mostrará la siguiente palabra almacenada en el archivo JSON.

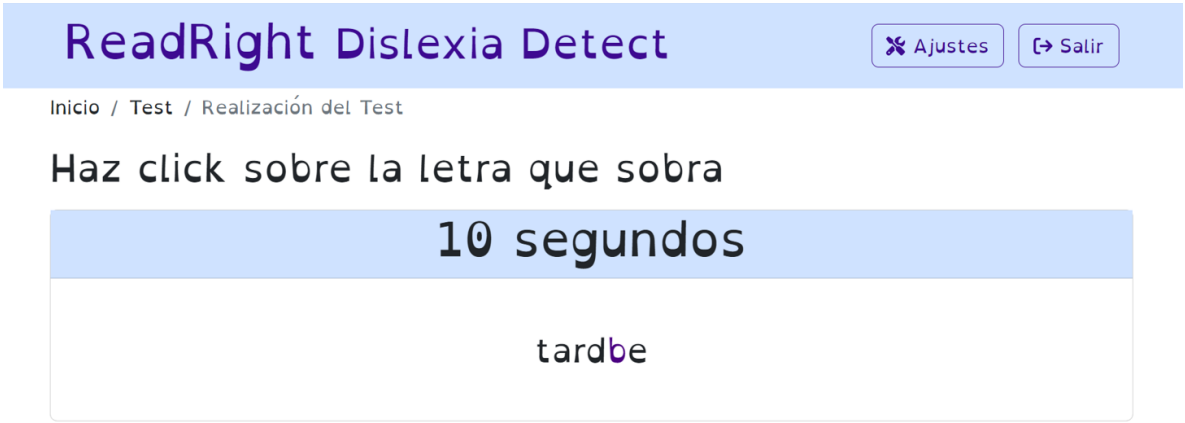


Figura D.6: Ejercicio 23

En el archivo JSON, los elementos están almacenados como pares clave:valor, donde el valor representa la letra que sobra. Para verificar si el usuario ha acertado, se comparará el valor del elemento pulsado con el valor almacenado en el JSON, sumando puntos a los aciertos si coinciden, o a los fallos si no lo hacen.

D.5. Ejercicios 24 y 25

En los ejercicios 24 y 25, el objetivo es encontrar la palabra que no tiene sentido en la frase. Todas las palabras son elementos clickeables que reaccionan cuando el cursor pasa por encima. El usuario deberá identificar cuál palabra no tiene sentido en el contexto de la oración, no porque esté mal escrita, sino porque no pertenece a la misma. Cada vez que se pulse una palabra, independientemente de si se ha encontrado la incorrecta o no, se mostrará la siguiente frase almacenada en el archivo JSON.

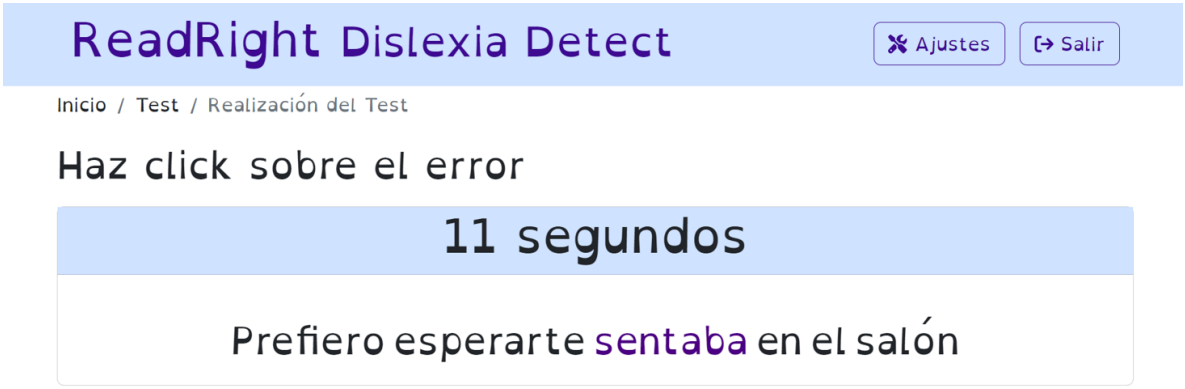


Figura D.7: Ejercicio 24

En el archivo JSON, los elementos están almacenados como pares clave:valor, donde el valor representa la palabra que no pertenece a la oración. Para verificar si el usuario ha acertado, se comparará el valor del elemento pulsado con el valor almacenado en el JSON, sumando puntos a los aciertos si coinciden, o a los fallos si no lo hacen.

D.6. Ejercicio 26

El ejercicio 26 se divide en dos partes. Primero, al usuario se le presenta una palabra mal escrita y debe identificar cuál letra está incorrecta. Cada letra es un elemento clickeable que reacciona cuando el cursor pasa por encima. Si el usuario pulsa una letra que no es la errónea, no pasará nada. Solo cuando haya identificado el error, aparecerá un conjunto de letras en formato de botón en la parte inferior de la pantalla. A continuación, el usuario deberá pulsar la letra correcta con el objetivo de corregir la palabra que se muestra en pantalla.

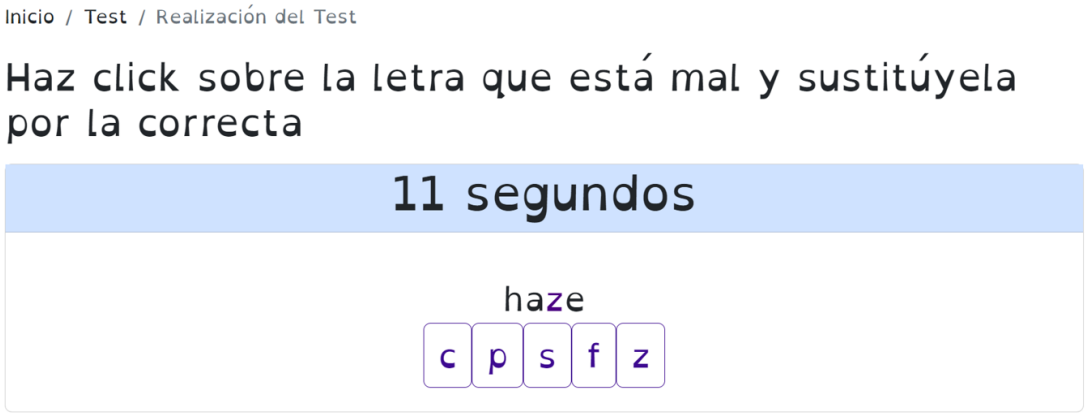


Figura D.8: Ejercicio 26

D.7. Ejercicios 27 y 28

En los ejercicios 27 y 28, se mostrarán en pantalla una serie de letras (Ejercicio 27) o sílabas (Ejercicio 28) en formato de botón que el usuario deberá ordenar para formar una palabra con sentido. Cada vez que se pulse un botón, en la parte superior se mostrará la palabra que se está formando. Cuando se pulse el último botón, se pasará al siguiente elemento almacenado en el archivo JSON.

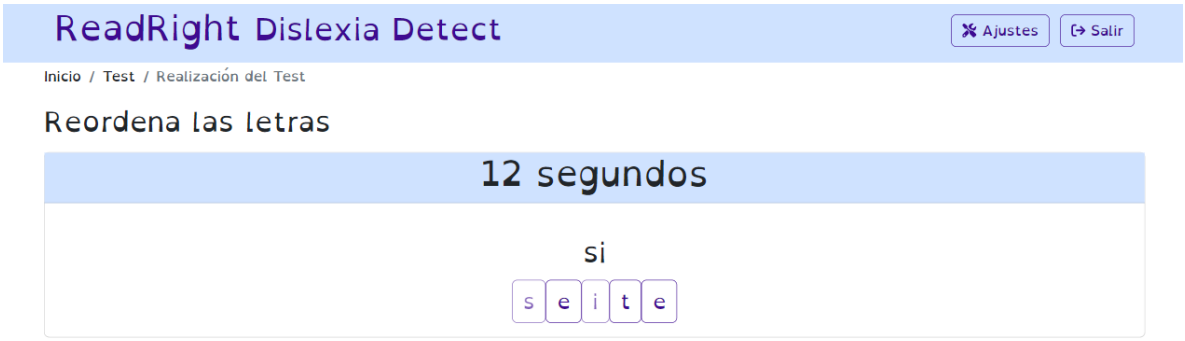


Figura D.9: Ejercicio 27

Una vez pulsada una letra, ya no se puede borrar y, si el usuario se ha equivocado, no podrá rectificarlo.

D.8. Ejercicio 29

En el ejercicio 29, se muestra una frase sin espacios que el usuario debe separar para que tenga sentido. Deberá reescribir la frase en el campo proporcionado y, cuando considere que ha terminado, pulsar el botón "Fin". Así, se le ofrecerá la siguiente frase, y podrá repetir el proceso tantas veces como le permita el tiempo.

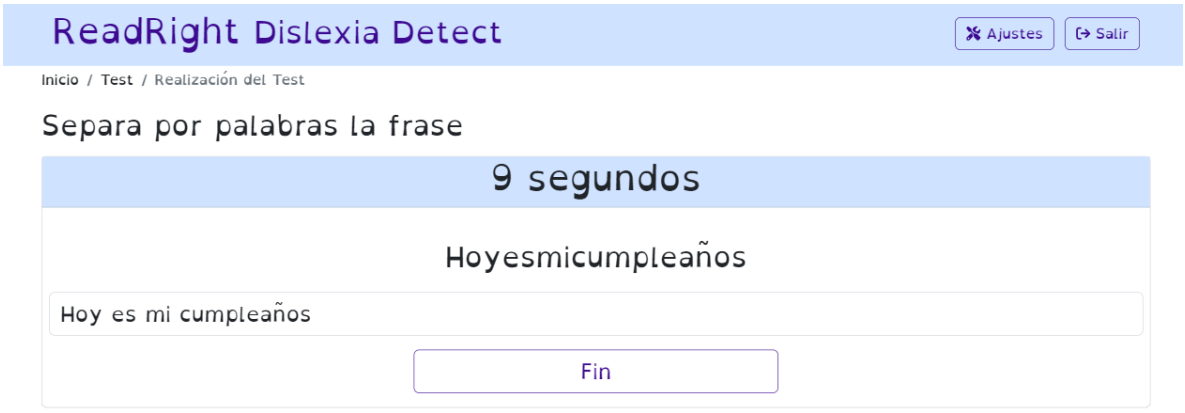


Figura D.10: Ejercicio 29

Para contabilizar el número de aciertos realizados, se comparará cada palabra que haya introducido el usuario (diferenciada por espacios) con las almacenadas en el archivo JSON. Si coinciden todas las

palabras que componen la frase, se sumará un acierto; en caso contrario, se sumará un fallo. Los clicks corresponden al número de palabras escritas por el usuario.

D.9. Ejercicio 30

El ejercicio 30 consta de dos partes. Primero, se mostrará durante tres segundos una secuencia de letras. Cuando haya expirado el tiempo, aparecerá un campo vacío donde el usuario podrá introducir la serie que acaba de ver. Este proceso se repetirá cuatro veces con cuatro series distintas, aumentando el nivel de dificultad progresivamente.

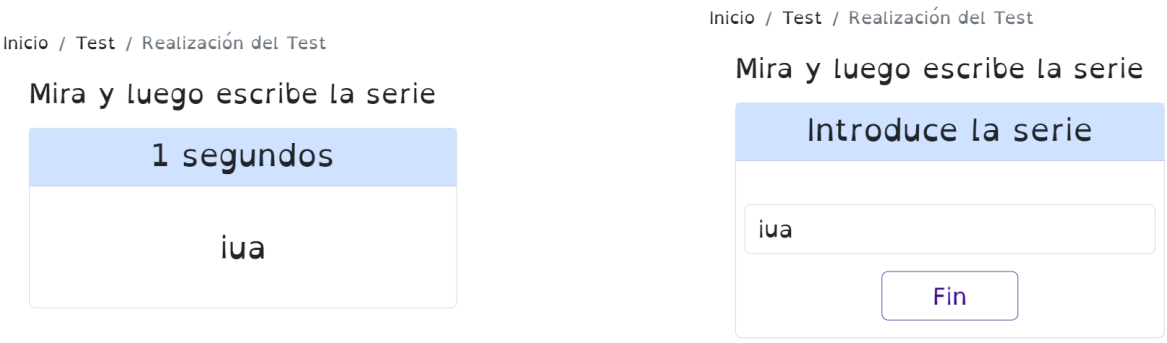


Figura D.11: Ejercicio 30

D.10. Ejercicios 31 y 32

Finalmente, en los ejercicios 31 y 32, el usuario deberá escribir la palabra escuchada. Se proporciona el botón "Escuchar audio", que el usuario deberá pulsar para poder escuchar la palabra que debe escribir en el campo situado en el cuerpo del ejercicio. Solo se puede escuchar dicho audio una vez. Tras escucharlo, el botón se desactiva. Cuando considere que ha terminado de introducir la palabra, deberá pulsar el botón "Fin" con el objetivo de acceder a la siguiente palabra. Este proceso se repetirá cuatro veces para cada ejercicio, mostrando cuatro palabras distintas en cada ocasión.

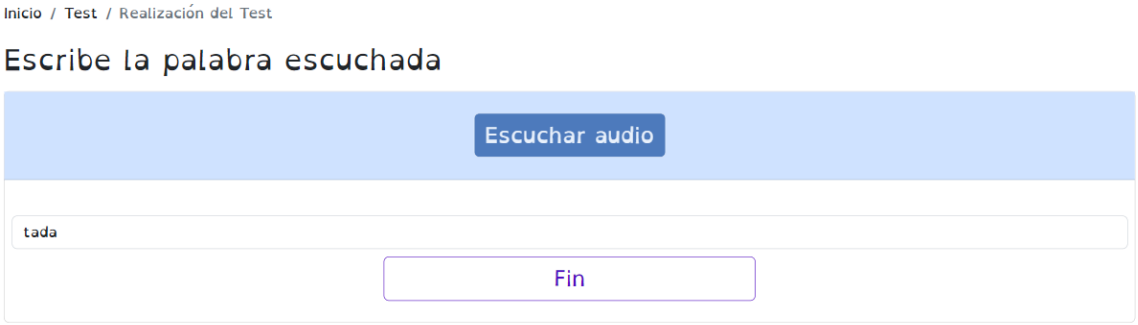


Figura D.12: Ejercicio 32

ANEXO D. TEST DE DETECCIÓN DE DISLEXIA

A excepción de los últimos tres ejercicios (29, 30 y 31), todos tienen una duración de quince segundos para completarse. En cada uno de ellos, se contabilizará el número de aciertos, fallos y clicks, parámetros que se almacenarán en la base de datos cada vez que se realice un ejercicio.

Anexo E

Ejercicios de entrenamiento

A su vez, la página web ofrece una serie de ejercicios que una persona con dislexia, ya sea diagnosticada por los métodos tradicionales o a través del modelo de detección de dislexia, puede realizar con el objetivo de entrenar aquellos procesos cognitivos que se hayan visto afectados a causa de la dislexia.

De la misma forma que en los ejercicios que componían el test de detección de dislexia, la lógica de los ejercicios se ha implementado mediante JavaScript, empleando un archivo JSON. Se han incluido los títulos e instrucciones que debe seguir el usuario, así como los objetos que componen el ejercicio, los cuales son necesarios para su diseño.

E.1. Juego del Ahorcado

El primer ejercicio que se muestra es el juego del ahorcado. En el archivo JSON, se cuenta con una lista de palabras, la cual se recorre de forma progresiva. El objetivo es que el usuario adivine qué palabra falta en cada ronda, mostrando simplemente el número de letras que tiene cada una, representadas mediante espacios con el símbolo "_". Se mostrarán tantos huecos en blanco como letras tenga la palabra a adivinar, así como la representación gráfica de los intentos que ha realizado el usuario. En la parte inferior de la imagen, se proporciona un campo de entrada donde el usuario podrá introducir las letras junto a un botón para comprobar si forman parte de la palabra o no. Las letras que el usuario haya introducido y que no conformen la palabra se mostrarán agrupadas en la parte inferior del contenedor.

Se recoge la letra enviada por el usuario y se compara con las letras de la palabra. En caso de que coincida con alguna letra, se sustituirá el espacio en blanco (_) por la letra. Si por el contrario no coincide, se añadirá la letra al conjunto de letras erróneas y se actualizará la imagen, completando la figura del ahorcado progresivamente.

El usuario tiene hasta seis intentos para adivinar la palabra. Cuando se haya alcanzado el número máximo de intentos, si aún no ha descubierto cual era la palabra, se mostrará en pantalla un mensaje indicando que se han consumido todos los intentos y cuál era la palabra a adivinar. En caso de que se acierte la palabra antes de que se acaben los intentos, aparecerá en pantalla un mensaje distinto indicando que ha ganado.

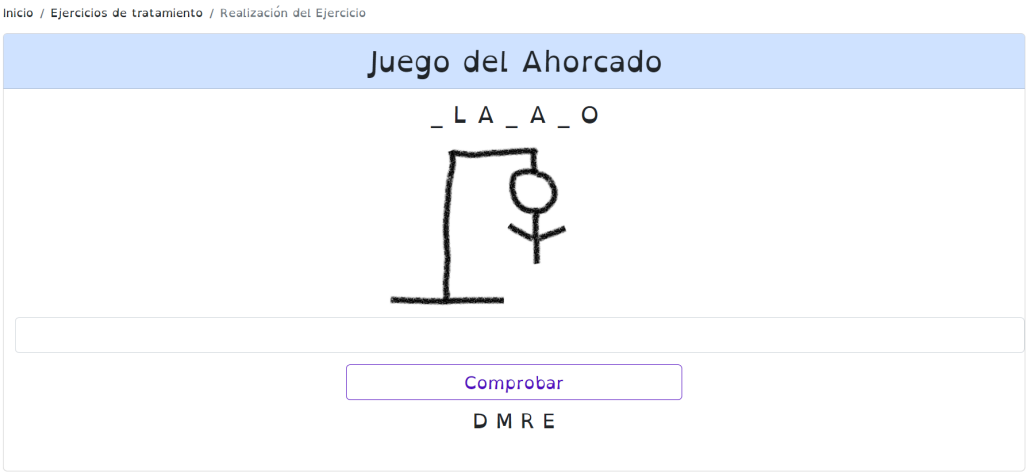


Figura E.1: Juego del Ahorcado

E.2. Juego de la Memoria

El juego de la memoria es una actividad que estimula la memoria de trabajo, fomenta la concentración y la atención, y ayuda con la búsqueda de patrones. Por ello, se ha decidido incorporarlo a la selección de ejercicios de tratamiento.

Se muestra una cuadrícula 4 x 4, formada por dieciséis imágenes dadas la vuelta. El usuario podrá levantar de dos en dos y tendrá que encontrar las ocho parejas con el menor número de intentos posible. Levantando de dos en dos, tendrá que identificar las parejas hasta que no quede ninguna carta boca a bajo. Si las dos imágenes que levanta coinciden, se mantendrán visibles. De lo contrario, tendrá dos segundos para memorizar las imágenes y se volverán a dar la vuelta.

Figura E.2: Juego de la Memoria

E.3. Tres en raya

Se ha incorporado el juego Tres en Raya a la aplicación, ya que, aunque es un juego extensamente conocido y popular, ayuda a estimular la atención y concentración, así como a reforzar las habilidades visoespaciales[1] y la planificación.

Se mostrará al usuario una cuadrícula de tres por tres que representa el tablero. El usuario jugará con las x's y el ordenador con las o's. Por defecto, se ha establecido que el usuario inicie la partida. El juego es interactivo, de modo que cada vez que un participante elija una posición, será el turno del siguiente. El juego finalizará cuando no se pueda rellenar más el tablero, y se mostrará la opción de continuar jugando o abandonar el juego.

Se ha incorporado lógica a los movimientos del ordenador, lo que hace que el usuario deba pensar en qué posición colocar su ficha. Esto contribuye a potenciar el pensamiento secuencial y la planificación durante el juego.

Figura E.3: Tres en raya

E.4. Twister

Con el objetivo de reforzar las habilidades espaciales, se ha incluido una adaptación del juego Twister. Se mostrará al usuario un espacio de trabajo dividido en tres secciones: las dos columnas externas, que contienen una imagen levantando la mano derecha situada a la derecha y una imagen levantando la mano izquierda situada a la izquierda, y el tablero situado en el centro, compuesto por los círculos de colores característicos del juego. Estos contenedores alojarán copias de las imágenes correspondientes a la derecha o a la izquierda, que el usuario arrastrará en función de las instrucciones recibidas.

En la parte superior del tablero, se ha situado un botón con el nombre "Girar". Este botón mostrará un mensaje indicando al usuario qué color y qué figura debe mover. Esta elección se realizará de forma completamente aleatoria. El usuario arrastrará la figura situada a la derecha o a la izquierda, según lo especificado, a uno de los círculos que coincida con el color. Si acierta, una réplica de la imagen se mantendrá en la posición elegida, de manera que el usuario sepa qué contenedores están ocupados. Si se equivoca, no se añadirá nada.

[1] Capacidad de decir dónde están los objetos en el espacio

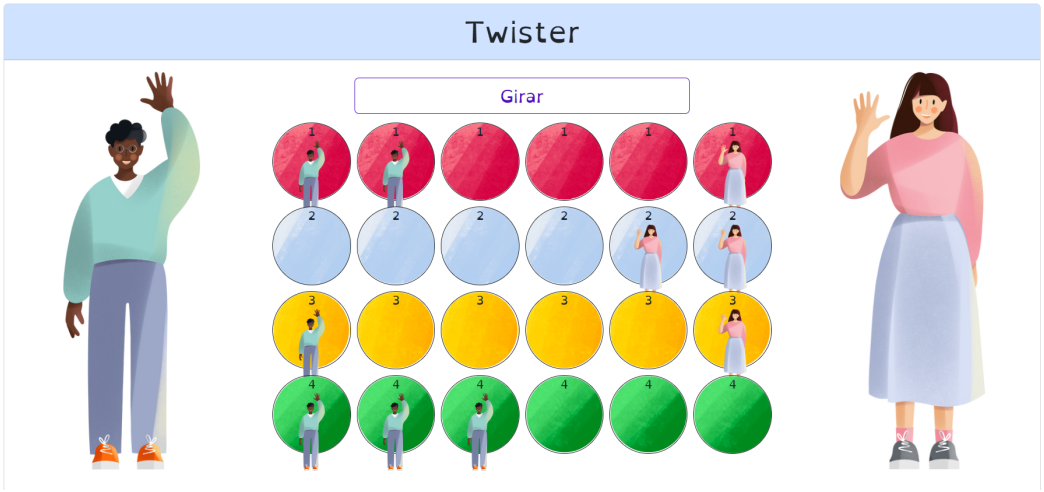


Figura E.4: Fin de la partida del Twister

Cada ronda se presentará al usuario tal y como se muestra en la Figuras E.5 y E.6:

Figura E.5: Instrucciones del Twister

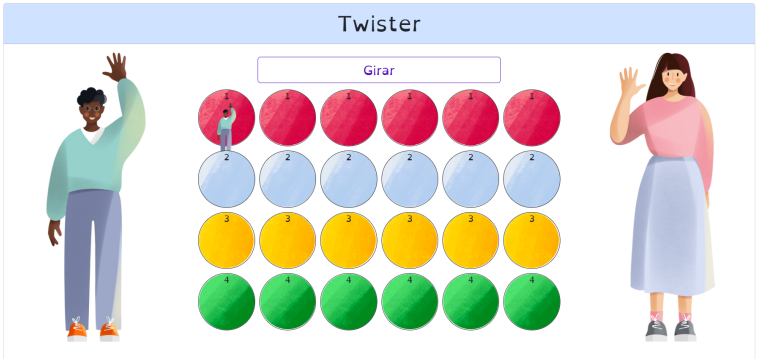


Figura E.6: Movimiento realizado en el Twister

El usuario podrá pulsar el botón "Girar"hasta diez veces. Una vez llegue a la décima, se mostrará un mensaje en pantalla para indicarle que la partida ha finalizado, permitiéndole empezar una nueva partida.

E.5. Asociar imágenes con la palabra correspondiente

El siguiente ejercicio consta de tres rondas, y en cada una se mostrarán cuatro imágenes distintas junto con una palabra que corresponda únicamente a una de ellas. La palabra que el usuario debe procesar se seleccionará de manera aleatoria en cada ronda. El usuario tiene que ser capaz de entender cada palabra e identificar lo que representa, pulsando la imagen correspondiente.

El usuario tendrá que pulsar la imagen que considere que se asocia con la palabra mostrada. Una vez pulse una imagen, se comparará la palabra asociada a la imagen pulsada con la palabra que se ha mostrado en pantalla. Si coinciden, el usuario habrá acertado y se mostrará un mensaje en pantalla indicándolo.

Figura E.7: Asociar imágenes con la palabra - Ronda 1

Figura E.8: Asociar imágenes con la palabra - Ronda 2

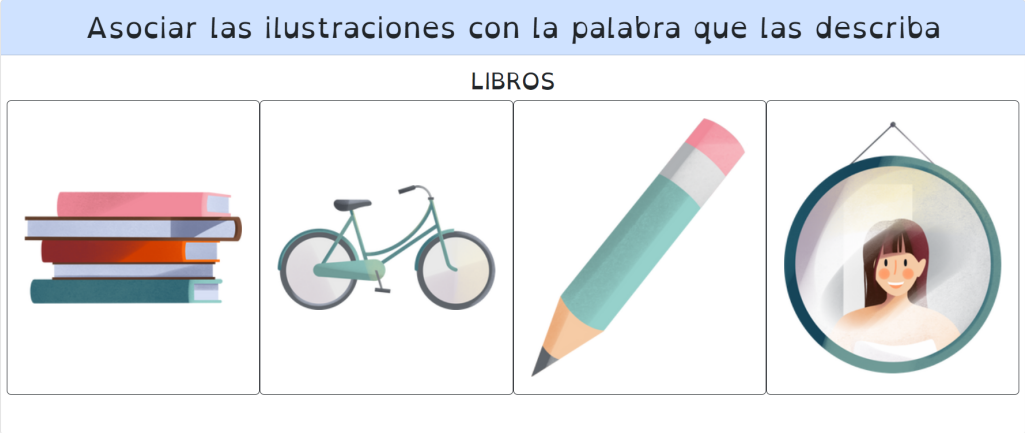


Figura E.9: Asociar imágenes con la palabra - Ronda 3

E.6. Reordenar las letras para formar una palabra

El último ejercicio que se ha incluido como actividades de entrenamiento se ha obtenido directamente del test de detección de dislexia. Corresponde a los ejercicios 27 y 28 del test.

Al usuario se le mostrarán una serie de letras desordenadas. Tendrá que pulsar en orden cada letra con el objetivo de reordenarlas y formar la palabra correcta.

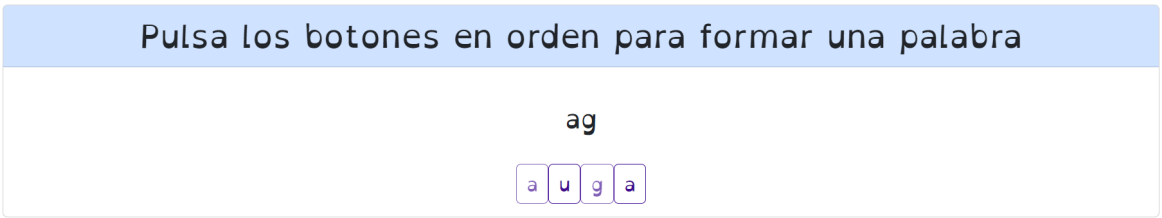
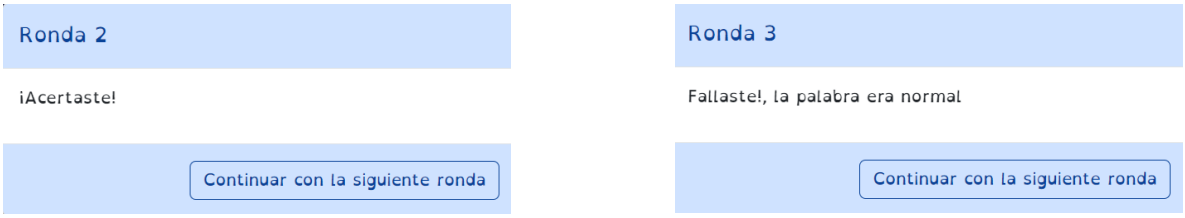


Figura E.10: Reordenar las letras para formar una palabra

Se comprueba en cada ronda si la palabra formada es la correcta o no. En caso de que se haya fallado, se informa de cual era la palabra corregida y se continúa con la siguiente ronda.



(a) Palabra acertada (b) Palabra fallada

Figura E.11: Mensajes Informativos en el ejercicio de reordenamiento de letras