

Brain Wave

-Improving the future-

Proyecto final BD10 KeepCoding 2023

Pablo Rilo Mariñas
Belisa Salcedo Viñuales
Emmanuel Alejandro Alma
Daniel Sarabia Torres
Laura Castaño Pujalte

1. Introducción	4
1.1. Model Business Canvas	5
2. Datos	5
2.1. Dataset	5
2.1.1. Primer Dataset descartado	5
2.1.2. Dataset Final	6
2.2. Datos de contraste	9
3. Arquitectura	21
3.1. Diseño	21
3.1.1.1. Premisas para el desarrollo de la arquitectura	21
3.2. Desarrollo	23
3.2.1. API	24
3.2.1.1. Introducción	24
3.2.1.2. Estructura del proyecto	24
3.2.1.3. Modelos de datos:	25
3.2.1.4. Autentificación	29
3.2.1.5. Endpoints	29
3.2.1.6. Tecnologías	33
3.2.1.7. Dependencias	34
3.2.2. Web	35
3.2.2.1. Introducción	35
3.2.2.2. Estructura del proyecto	35
3.2.2.3. Arquitectura	37
3.2.2.4. Tecnologías	37
3.2.3. Modelo	38
4. Visualización BI	39
4.1. KPIs	39
4.2. Gráficas	41
5. Resolución de la problemática	43
5.1. Análisis de datos	43
5.2. Preprocesado	45
5.3. Modelado	45
5.3.1. Explicación del Modelo	45
5.3.2. Evaluación del modelo	49
6. Despliegue en la nube	53
6.1. Información general	53
6.2. Estructura de carpetas	53
6.3. Pasos generales del despliegue	54
6.3.1. procesamiento de datos	54
6.3.2. Entrenando al modelo (training.py) (sin implementar)	55
6.3.3. Predicciones (predict.py) (sin implementar)	55
6.3.4. Agrupamiento de datasets y reentrenamiento (retraining.py) (sin implementar)	55

56	
6.4. Tecnologías	56
7. Presentación de resultados	58
7.1. Suposiciones iniciales. A completar	58
7.2. ¿Cuáles nos han sido válidas? ¿Cuáles no? A completar	58
7.3. Métricas seleccionadas y por qué A completar	58
7.4. Arquitectura elegida. ¿Ha sido la definitiva?	59
7.5. Métodos elegidos. ¿Cuáles han sido los mejores? A completar	62
7.6. Retrospectiva ¿Qué haríamos igual?¿Qué cambiaríamos?	62
7.7. Información obtenida del dataset	62
7.8. Conclusiones	63
8. Demo	63



1. Introducción

En el momento de plantear el tema de nuestro proyecto, buscamos uno que tuviera una relevancia social actual y que pudiera ser útil para ayudar a nuestra sociedad. Después de realizar diversas búsquedas de conjuntos de datos relacionados con estos términos, nos dimos cuenta de que el bullying es uno de los principales problemas a nivel mundial, y que no había mucho trabajo realizado en la detección de bullying mediante técnicas de inteligencia artificial.

Decidimos, por lo tanto, elegir este tema para nuestro proyecto y comenzamos a buscar un conjunto de datos que pudiera servir como base para nuestro estudio. Encontramos un conjunto de datos en Kaggle que pensamos podría ser adecuado, sin embargo, pronto nos dimos cuenta de que no era lo suficientemente completo para nuestros propósitos. Continuamos investigando y finalmente encontramos un dataset más completo y adecuado para nuestro trabajo

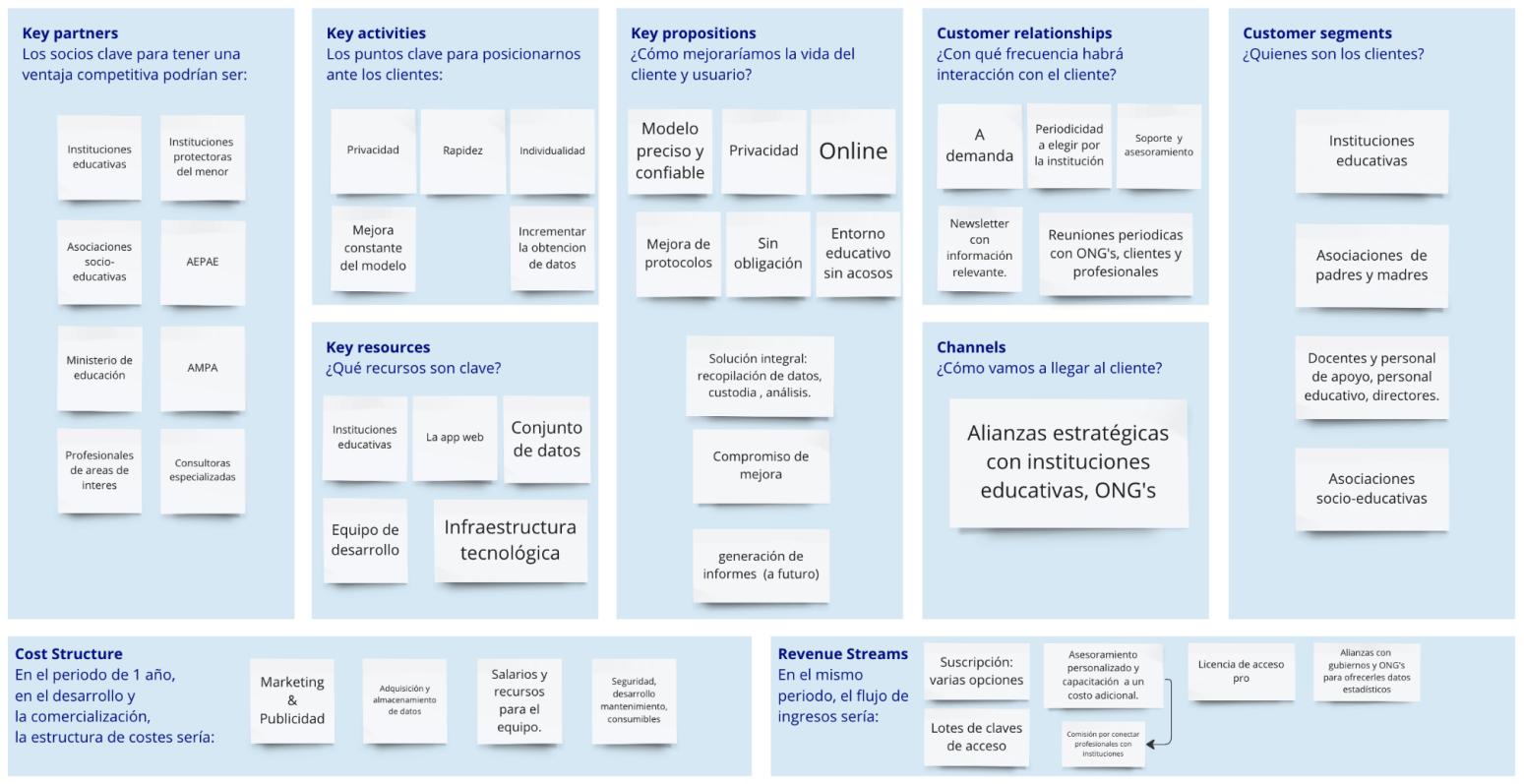
El conjunto de datos utilizado para el estudio ha sido obtenido a través de un formulario dirigido a estudiantes en Argentina. Los datos obtenidos han sido utilizados como base para el análisis y la implementación de modelos de inteligencia artificial para la detección de casos de bullying. La organización que ha facilitado los datos utilizados en el presente estudio es la Organización Mundial de la Salud (OMS), la cual ha proporcionado acceso a los datos a través de su página web oficial en el siguiente enlace:

https://extranet.who.int/ncdsmicrodata/index.php/catalog/866/data-dictionary/F5?file_name=National

Dichos datos corresponden a información recopilada por la OMS a nivel nacional en Argentina, y han sido utilizados como base para el análisis realizado en este proyecto de investigación en el campo de la inteligencia artificial. Cabe destacar que, a pesar de la búsqueda exhaustiva llevada a cabo por el equipo, no se ha encontrado ningún dataset disponible con información real y útil para realizar un proyecto de esta índole con datos de España.

1.1. Model Business Canvas

The Business Model Canvas



2. Datos

2.1. Dataset

2.1.1. Primer Dataset descartado

El primer dataset que evaluamos, obtenido de Kaggle, consiste en un conjunto de datos de 56,981 registros y 18 características. Las columnas de este dataset incluyen información como, edad, género, si ha sido atacado físicamente, si ha peleado físicamente, si se ha sentido solo, si tiene amigos cercanos, si ha faltado a la escuela sin permiso, si otros estudiantes han sido amables y serviciales, si los padres entienden los problemas del estudiante, si se ha sentido solo la mayor parte del tiempo o siempre, si ha faltado a clases o escuela sin permiso, y si el estudiante tenía bajo peso, sobrepeso u obesidad. Las variables objetivo de este conjunto de datos serían: bullying en la propiedad escolar en los últimos 12 meses, bullying fuera de la propiedad escolar en los últimos 12 meses y ciberacoso en los últimos 12 meses.

Después de realizar una limpieza de datos en el dataset original, se procedió a eliminar aquellas filas que contenían valores nulos. Como resultado, se obtuvo un nuevo conjunto de datos que consta de 32,938 filas y 18 columnas. Cabe destacar que se eliminaron todas aquellas instancias que

presentaban algún valor faltante, con el objetivo de trabajar únicamente con información completa y de mayor calidad en el análisis posterior. Al eliminar las filas con datos nulos todavía tenemos una cantidad considerable de filas para realizar un estudio completo, lo que nos permite tener un dataset más limpio y apto para realizar un análisis más preciso. Es importante tener en cuenta que la eliminación de filas con datos nulos puede afectar la representatividad del dataset y, por lo tanto, es necesario evaluar si la muestra resultante es aún representativa de la población de interés.

A pesar de que se cuenta con una cantidad significativa de filas tras realizar la limpieza de datos, el número de atributos disponibles en el primer dataset obtenido es insuficiente para poder generar modelos de inteligencia artificial eficaces en la detección del bullying. Debido a esta limitación, los modelos planteados utilizando este dataset no han arrojado resultados satisfactorios en el contexto de nuestro proyecto.

2.1.2. Dataset Final

El segundo conjunto de datos consta de 56,981 filas y 155 columnas. Se ha llevado a cabo una limpieza de datos, eliminando inicialmente columnas que contenían información duplicada, columnas que no se proporcionaron en la página web y columnas con más del 37% de valores nulos. Después de la limpieza, se seleccionaron 51 columnas que incluyen información sobre la edad, género, altura, peso, actividad física, consumo de alcohol, drogas y tabaco, entre otras. Al eliminar las filas que contenían valores nulos, el conjunto de datos se redujo a 19,468 filas.

Después de realizar esta limpieza nos quedamos con estas columnas:

- q1 Custom Age
- q2 Sex
- q3 In what grade are you
- q4 How tall are you
- q5 How much do you weigh
- q6 How often went hungry
- q10 Fast food eating
- q15 Physically attacked
- q16 Physical fighting
- q17 Seriously injured
- q18 Serious injury type
- q19 Serious injury cause

q22 Felt lonely
q23 Could not sleep
q24 Considered suicide
q25 Made a suicide plan
q26 Attempted suicide
q27 Close friends
q28 Initiation of cigarette use
q29 Current cigarette use
q34 Initiation of alcohol use
q35 Current alcohol use
q36 Drank 2+ drinks
q37 Source of alcohol
q38 Really drunk
q39 Trouble from drinking
q40 Initiation of drug use
q41 Ever marijuana use
q42 Current marijuana use
q43 Amphetamine or methamphetamine use
q44 Ever sexual intercourse
q45 Age first had sex
q46 Number of sex partners
q47 Condom use
q48 Birth control used
q49 Physical activity past 7 days
q50 Walk or bike to school
q51 PE attendance
q52 Sitting activities
q53 Miss school no permission
q54 Other students kind and helpful
q55 Parents check homework

q56 Parents understand problems
q57 Parents know about free time
q58 Parents go through their things
qn66 Bullied on school property in past 12 months
qn67 Bullied not on school property in past 12 months
qn68 Cyber bullied in past 12 months
qnunwtg
qnowtg Were overweight
qnobeseg Were obese

El documento "Anexo1.pdf" contiene información detallada sobre cada pregunta y las posibles respuestas del cuestionario utilizado en el estudio. Este anexo puede ser consultado para obtener una comprensión más completa de los datos recopilados en el cuestionario con el que se ha creado el dataset utilizado en el proyecto.
Hemos realizado diferentes modelos poniendo como variable objetivo diferentes variables.

a) Como variable objetivo:

qn66 Bullied on school property in past 12 months
qn67 Bullied not on school property in past 12 months
qn68 Cyber bullied in past 12 months

b) Como variable objetivo:

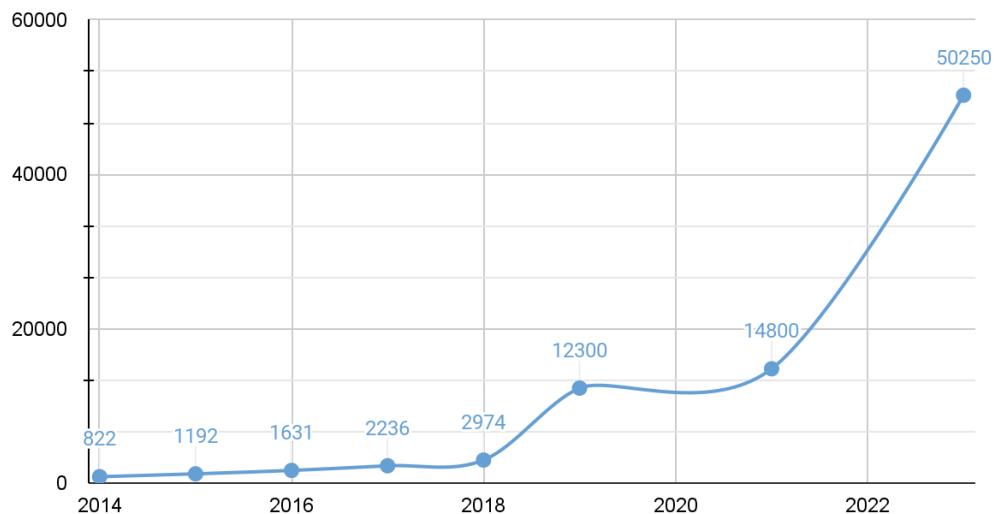
qn66 Bullied on school property in past 12 months
qn67 Bullied not on school property in past 12 months

c) Como variable objetivo:

q15 Physically attacked

2.2. Datos de contraste

Evolución de casos de bullying en Argentina.



45.4% de los niños perciben que el profesor no hace nada.

61.7% sienten que su centro no se implica.

50% considera que sus compañeros no hacen algo para evitarlo.

Acoso escolar y sus huellas.

El acoso escolar causa problemas psicológicos en el 90% de los niños que lo sufren, según un estudio realizado por la Fundación Mutua Madrileña y la Fundación ANAR, los hechos de acoso se han vuelto más violentos, intensos y frecuentes.

Aunque el acoso de baja intensidad ha disminuido, persiste el acoso más grave, perpetrado por agresores persistentes y crueles.

El perfil de las víctimas se mantiene constante en los últimos años, con una ligera igualdad de género en el acoso escolar y una prevalencia de mujeres en el ciberacoso.

El acoso tiene un impacto duradero en las víctimas, en el 97% de los casos y una duración prolongada de más de un año.

Las formas de acoso más comunes incluyen insultos, ofensas verbales, agresiones físicas leves y aislamiento.

Se ha observado un aumento en la violencia física y una mayor incidencia de acoso en aulas, durante los recreos y en los cambios de clase.

La violencia y la frecuencia del acoso aumentan a medida que persiste en el tiempo, especialmente en el caso del ciberbullying.

Los agresores suelen justificar sus acciones por:
las características de las víctimas
Comportamiento
Rendimiento escolar
Habilidad deportiva
Diversión.

Aunque algunos profesores no reaccionaron ante la violencia, aquellos que sí lo hicieron mostraron una actitud más activa y comunicativa con los agresores, las víctimas y sus familias.

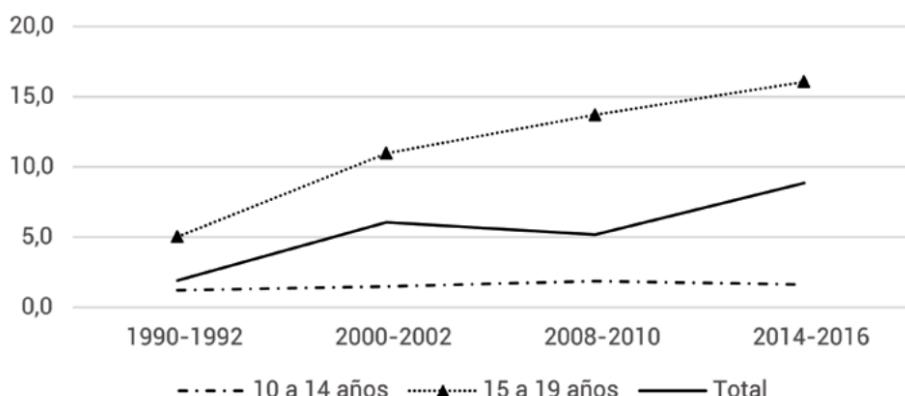
Las consecuencias del acoso son graves y van desde problemas psicológicos, como síntomas depresivos y ansiedad, hasta autolesiones e ideas suicidas.

Se observa una tendencia descendente en las consecuencias más graves, pero aún persisten.

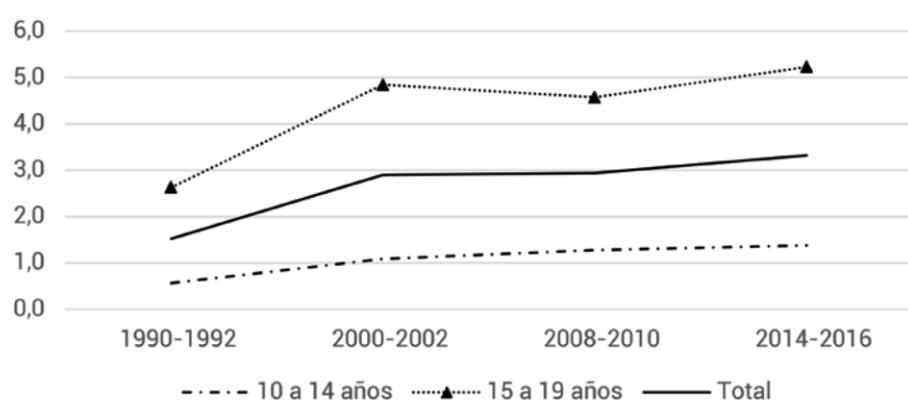
Para finalizar este párrafo es importante tener en cuenta que el entorno familiar de las víctimas suele experimentar problemas psicológicos relacionados con el acoso.

Mortalidad.

Tasa de suicidio adolescente cada 100.000. Varones. Argentina. 1990-2016



Tasa de suicidio adolescente cada 100.000. Mujeres. Argentina. 1990-2016



Los datos en los certificados de defunción en la Argentina únicamente incluyen información acerca del nivel educativo, por eso se usó esta como variable indicadora del nivel socioeconómico.

Del análisis surge que los adolescentes varones con menor nivel educativo tienen aproximadamente tres veces más posibilidades de cometer un suicidio que los adolescentes varones con un nivel educativo de secundaria completa o más.

En el caso de las mujeres adolescentes con nivel educativo hasta primario completo, tienen aproximadamente 1.7 más posibilidades de cometer suicidio que sus pares con un mayor nivel educativo.

En la Argentina, los suicidios constituyen la segunda causa de muerte en la franja de 10 a 19 años

En el grupo de 15 a 19 años, la mortalidad es más elevada, alcanzando una tasa de 12,7 suicidios cada 100.000 habitantes, siendo la tasa en los varones 18,2 y en las mujeres 5,9

Desde principios de la década de 1990 hasta la actualidad la mortalidad por suicidio en adolescentes se triplicó considerando el conjunto del país

Ahora, hablemos de las principales causas segun sexo:

Mortalidad adolescente por suicidio según tipo de suicidio y sexo. Argentina. 2012-2016.

Tipo de suicidio	Sexo	
	% de varones	% de mujeres
Envenenamiento	0,6	2,8
Ahorcamiento	87,8	87,8
Disparo de arma de fuego	9,3	4,7
Saltar desde un lugar elevado	1,7	3,5
Otros especificados	0,6	1,2
	100,0	100,0
Total de casos	(1596)	(599)

Aunque predomina el ámbito urbano con un 60%, más de un 25% de los casos corresponden a adolescentes residentes en áreas periurbanas, es decir localidades entre 4000 y 30.000 habitantes, que por sus características productivas y geográficas se encuentran en una frontera poco precisa en relación con el ámbito rural no disperso o urbano.

Casos de suicidios consumados y tentativas según ámbito de residencia

Ámbito	Suicidios consumados	Tentativas	Total
Urbano	16	17	33
Periurbano	5	10	15
Rural	6	1	7
Total	27	28	55

NSE : Nivel socioeconómico

Para crear estos 3 grupos se unió el tipo de vivienda, máximo nivel educativo de padres o responsables y la ocupación de estos.

Casos de suicidios consumados y tentativas según NSE

NSE	Suicidios consumados	Tentativas	Total
Baja	14	10	24
Media baja	2	8	10
Media	11	10	21
Total	27	28	55

Casos de suicidios consumados y tentativas según sexo

Sexo	Suicidios consumados	Tentativas	Total
Femenino	12	21	33
Masculino	15	6	21
Transsexual	-	1	1
Total	27	28	55

Según las estadísticas de mortalidad mencionadas anteriormente, se observa que hay más hombres fallecidos que mujeres.

Sin embargo, las tentativas de suicidio muestran lo contrario. En general, los datos de mortalidad indican que los hombres tienden a utilizar métodos más letales que las mujeres.

Esto se refleja en la muestra seleccionada para este estudio, donde el ahorcamiento o sofocación es el método predominante tanto para hombres como para mujeres, con más del 80% de los casos.

Métodos empleados en los casos de suicidios consumados

Método empleado	Mujeres	Varones	Total
Ahorcamiento	11	11	22
Uso de arma de fuego	1	1	2
Asfixia	-	1	1
Salto de lugares elevados	-	1	1
Ahogamiento voluntario en río	-	1	1
Total	12	15	27

Los métodos empleados en los intentos de suicidio suelen ser menos letales que en los suicidios consumados, tal como sucede en la muestra seleccionada: el uso de pastillas u otro tipo de sustancias ingeridas alcanza en conjunto cerca del 42% de los casos de tentativas y los cortes con objetos cortopunzantes cerca del 29%.

Métodos empleados en los casos de tentativas

Método empleado	Mujeres	Varones	Transexuales	Total
Pastillas/otras sustancias	10	1	1	12
Objeto cortante	6	2	-	8
Ahorcamiento	2	2	-	4
Salto de lugares elevados	3	1	-	4
Total	21	6	1	28

Un dato que llama la atención es que, a la inversa de lo que muchas veces se cree, la presencia de intentos previos de suicidios son indicadores de riesgo, sin embargo 8 de cada 10 suicidios no tenían registro de intento previo.

Tipos ideales

Son construcciones que los profesionales realizan para sintetizar las situaciones vividas, el entorno familiar, educativo, y comunitario en el que desarrollaba su vida el adolescente

Estos tipos no pueden generalizar la población total, pero sirven como herramienta de estudios y orientación para comprender experiencias pasadas e intentar ayudar a futuras víctimas de manera temprana.

Tipología de los casos de suicidios consumados	
1. Adolescentes que han atravesado situaciones de inexistencia o pérdida de soporte.	
1.1. Adolescentes sin contención familiar con los que las instituciones intervienen fracasaron.	
1.2. Adolescentes que sufrieron la pérdida de una relación afectiva que constituyó su principal soporte.	
2. Adolescentes que sufrieron o temen sufrir desfasajes entre sus expectativas y sus logros.	
2.1. Adolescentes que experimentan una aguda sensación de fracaso frente a sus propias expectativas de logro y/o que consideran que no cumplen con las expectativas de sus familias o de otras instituciones como los mandatos religiosos.	
2.2. Adolescentes que experimentan temor al fracaso frente a inminentes pruebas de paso a la juventud/adulz, por ejemplo, la terminación del secundario y el ingreso a estudios terciarios/universitarios o al mundo laboral.	
3. Adolescentes que interiorizaron esquemas valorativos rígidos, que no admiten ser confrontados por situaciones que implican valores contrarios, y por lo tanto son vividas como hechos traumáticos no procesables.	
4. Adolescentes en los que existe un componente de enfermedad mental evidente, que no llegaron o no fueron debidamente atendidos por las instituciones de salud mental.	
	1. Adolescentes que atravesaron situaciones de inexistencia o de pérdida de soportes, paliadas por adultos o instituciones que desempeñaron un rol protector.
	1.1. Adolescentes con limitada contención familiar, con los que las instituciones intervienen cumplieron aunque sea parcialmente sus cometidos como protectoras.
	1.2. Adolescentes que sufrieron la pérdida de una relación afectiva que constituyó su principal soporte, con los que una figura adulta o las instituciones cumplieron luego un rol protector.
	2. Adolescentes que sufrieron o temen sufrir desfasajes entre sus expectativas y sus logros, pero con los que una figura o las instituciones intervienen cumplieron, aunque sea parcialmente sus cometidos como protector.
	2.1. Adolescentes que experimentan una sensación de fracaso frente a sus propias expectativas de logro y/o que consideran que no cumplen con las expectativas de sus familias o de otras instituciones como los mandatos religiosos, pero con los que las instituciones cumplieron, aunque sea parcialmente un rol protector.
	2.2. Adolescentes que experimentan temor al fracaso frente a inminentes pruebas de paso a la juventud/adulz, por ejemplo, la terminación del secundario y el ingreso a estudios terciarios/universitarios o al mundo laboral, pero con los que una figura o las instituciones cumplieron, aunque sea parcialmente un rol protector.
	3. Adolescentes que interiorizaron esquemas valorativos rígidos, que no admiten ser confrontados por situaciones que implican valores contrarios, y por lo tanto son vividas como hechos traumáticos no procesables, pero con los que las instituciones cumplieron, aunque sea parcialmente un rol protector.
	4. Adolescentes en los que existe un componente de enfermedad mental, con los que las instituciones cumplieron, aunque sea parcialmente un rol protector.

Atención Médica

Los servicios de salud por lo general desconocen o carecen de protocolos o de procedimientos que guíen las intervenciones de los profesionales.

Más de la mitad de los entrevistados desconocían la existencia de los Lineamientos para la Atención del Intento de Suicidio en Adolescentes.

La inexistencia de equipos interdisciplinarios con la presencia de psicólogos o psiquiatras en muchos servicios de atención, en especial en las guardias. Faltan espacios de reflexión sobre la práctica.

Muchas de las capacitaciones solo se dan en el marco de las actividades de pases de sala o en ateneos.

La escasez de redes institucionales que conozcan y difundan los recursos existentes, como por ejemplo las camas disponibles para internación, también dificulta la tarea en los servicios

Prevencion de suicidio.

Se recomienda un enfoque de trabajo en equipo que involucre a docentes, médicos, enfermeras, psicólogos y trabajadores sociales, en colaboración con organizaciones comunitarias.

Existen publicaciones y documentos de referencia a nivel nacional y provincial que brindan procedimientos y recomendaciones para el tratamiento de tentativas de suicidio y suicidios consumados en el entorno escolar.

En cuanto a la identificación de personas en riesgo, se pueden utilizar cuestionarios estandarizados como la ISO 30.

Tabla 1 Estadísticos descriptivos por dimensión de la versión original del ISO-30

Factor/Variable	M	DE	As	K	CD
Baja Autoestima					
BA1 Yo debo ser un soñador/a, ya que estoy siempre esperando cosas que no resultan	.60	.734	.781	-.749	.176
BA2 Mientras crecía me hicieron creer que la vida podría ser justa. Siento que me mintieron, ya que no es justo en absoluto	.63	.772	.748	-.933	.318
BA3 Tengo las cualidades personales que necesito para que me guíen hacia una vida feliz.	.36	.686	1.61	1.05	.321
BA4 Cuando veo a alguien que logró lo que yo no tengo, siento que es injusto.	.31	.60	1.79	2.01*	.258
BA5 Solía pensar que podía ser alguien especial, pero ahora veo que no es verdad.	.49	.727	1.13	-.201	.457
BA6 Nadie me amaría si realmente me conociese bien.	.51	.768	1.10	-.401	.309
Desesperanza					
DES1 Hay muchas posibilidades para mí de ser feliz en el futuro.	.16	.492	3.10	8.36*	.333
DES2 Mi vida se ha desarrollado mayormente en las direcciones que yo elegí.	.51	.752	1.08	-.385	.131
DES3 Cuando me pasa algo malo siento que mis esperanzas de una vida mejor son poco reales.	2.38	0.75	-.752	-.847	-.38
DES4 Aun cuando me siento sin esperanzas, sé que las cosas eventualmente pueden mejorar.	.28	0.63	2.04	2.62*	.404
DES5 Siento que tengo control sobre mi vida.	.48	.743	1.18	-.174	.132
DES6 Es posible que me convierta en la clase de persona que quiero ser.	.42	.724	1.37	.304	.285
Incapacidad para Afrontar Emociones					
IAE1 Generalmente pienso que an los peores sentimientos desaparecerán.	.57	.762	.908	-.694	.128
IAE2 Yo debería ser capaz de hacer que duren los buenos momentos, pero no puedo.	.74	.787	.505	-1.21	.247
IAE3 An cuando estoy muy enojado/a por algo, puedo forzarme a mí mismo a pensar claramente, si lo necesito.	.42	.697	1.36	.401	.154
IAE4 Cuando mi vida no transcurre fácilmente estoy dominado por una confusión de sentimientos.	.70	.767	.567	-1.09	.303
IAE5 Cuando tengo emociones fuertes mi cuerpo se siente fuera de control. Dominá mi carácter y no	.63	.803	.771	-1.01	.372
IAE6 Nunca sentí que estuviera a punto de hacerme pedazos (Quebrarme).	1.21	.833	-.419	-1.43	-.264
Ideación suicida					
IS1 Aquellas personas con las que me relaciono, no me necesitan en absoluto.	.43	.645	1.23	.334	.469
IS2 Creo que seré incapaz de encontrar suficiente coraje como para enfrentar la vida.	.61	.805	.827	-.96	.357
IS3 Para impedir que las cosas empeoren, creo que suicidarse es la solución.	.30	.62	1.88	2.17*	.483
IS4 Pienso en morirme como una forma de resolver todos mis problemas.	.32	.634	1.82	1.90	.578
IS5 Para no sentirme mal o solo (a), pienso que la solución es morirse.	.28	.595	1.99	2.69*	.658
SA6 Los buenos sentimientos que la gente tiene acerca de mí son un error.	.44	.703	1.29	.231	.483

M=media aritmética, DE=desviación estándar, As= coeficiente de asimetría, K=curtosis, CD=coeficiente de discriminación. * La variable no fue considerada para el AFC

Sin embargo, se ha cuestionado la existencia de indicadores específicos para detectar la posibilidad suicida, ya que estos indicadores pueden estar presentes en diversas problemáticas que afectan a los adolescentes.

Algunos argumentan que la aplicación de estos cuestionarios puede generar estigmatización y provocar el efecto contrario al pretendido.

Existen algunas alternativas que pueden ayudar a saber de manera general el ámbito de un grupo de alumnos, por ejemplo, el test de escala de desesperanza.

Ítems	Componentes		
	Falta de Motivación	Expectativas Futuras	Sentimientos respecto al Futuro
16. Como nunca consigo la que quiero no tiene sentido desear algo	.93		
9. Nada me ha salido bien hasta ahora y no hay razón para esperar algo mejor del futuro	.92		
20. No tiene sentido tratar de lograr lo que quiero, probablemente no lo voy a conseguir	.89		
17. Es muy difícil que yo encuentre alguna satisfacción en el futuro	.85		
11. Lo que puedo ver en mi futuro es desagradable más que agradable	.84		
12. No espero conseguir lo que realmente deseo	.81		
14. Las cosas nunca me salen como yo quiero que me salgan	.75		
2. Mejor me doy por vencido ya que nada puedo hacer para mejorar mi vida	.72		
7. Mi futuro parece oscuro	.60		
15. Tengo mucha fe en el futuro	.87		
5. Tengo el tiempo suficiente para lograr las cosas que quiero hacer	.80		
1. Miro hacia el futuro con esperanza y entusiasmo	.58		
18. El futuro me parece inseguro e incierto	-.52		
8. He tenido muy buena suerte en la vida y espero recibir más cosas buenas de la vida aún	.47		
10. Mis experiencias del pasado me han preparado bien para el futuro	.42		
3. Cuando las cosas andan mal, me ayuda saber que no será así para siempre	.61		
13. Pensando en el futuro espero sentirme más feliz de lo que me siento ahora	.60		
6. En el futuro, yo espero tener éxito en las cosas más importantes para mí	.54		
19. Puedo esperar más tiempos buenos que malos	.50		
4. No me puedo imaginar lo que será mi vida de aquí a diez años	.45		

Buenas prácticas.

Se han desarrollado talleres participativos en los que se busca canalizar los sentimientos de vulnerabilidad y reducir intentos suicidas.

Estos talleres se basan en una metodología que fomenta la construcción colectiva del conocimiento, utilizando la experiencia y la cultura local.

Se resalta la importancia de partir de la experiencia práctica de los participantes, reflexionar sobre ella y luego volver a la práctica enriquecida con el apoyo de un coordinador facilitador.

Un ejemplo en América Latina fue una estrategia de intervención en Colombia que tuvo como objetivo fomentar conductas protectoras y proporcionar herramientas a educadores y padres para trabajar con los factores de riesgo de suicidio en adolescentes.

Esta intervención demostró ser efectiva al aumentar el conocimiento de los padres y educadores, lo que facilitó la identificación y el tratamiento oportuno de adolescentes en riesgo.

Fuente: [UNICEF](#)

Testimonios de Docentes.

conclusión:

Discusión / conclusiones

A partir de los datos obtenidos en el [paper](#) de la 2da conferencia interdisciplinaria internacional celebrada en argentina, Buenos Aires en el 2016 se pueden extraer algunas conclusiones parciales.

- Los docentes no creen como factores causales del hostigamiento las familias de padres separados, familias numerosas o ensambladas
- Los docentes no creen que la competitividad en el área deportiva tenga que ver con el hostigamiento

- Los docentes creen que los padres son un factor muy importante en las medidas preventivas
- Los docentes creen que las medidas punitivas frente a la discriminación pueden ser algo útiles
- Los docentes no creen que el cambio de escuela resuelva el problema del hostigamiento

La autopercepción de los docentes en relación a sus intervenciones en el ámbito escolar.

Se observa que un grupo de docentes se autodenominó “las superpoderosas” en alusión a la paradoja entre la sensación de no contar con herramientas y lo que se espera de ellas: “que tengan respuesta para todo”.

Los docentes en su práctica encuentran dificultades para planificar y llevar adelante diferentes intervenciones.

Por un lado, refieren sentirse poco motivados, y señalan un grupo que difícilmente se involucra: “Existen muchos docentes que van a dar clases y nada más, no les interesa saber de sus alumnos”.

Por otro lado, les cuesta mucho el trabajo en conjunto, la planificación y ejecución de intervenciones requiere de un nivel de diálogo y decisiones compartidas muy difícil de lograr: “El individualismo hace que cada vez cueste más ponerse de acuerdo”.

Reflexionan acerca de limitaciones en el modo de posicionarse ante la participación de alumnos/as, asumen que una participación efectiva requeriría mayores niveles de confianza y apertura de parte de los adultos: “muchas veces en las escuelas se quiere habilitar la palabra del alumnado, pero de manera muy limitada. Cuando se les pide opinión los docentes terminan diciéndoles que lo que piensan está mal porque no tienen edad para opinar”.

Otro tópico al que aluden los docentes es la falta del apoyo de los directivos para realizar intervenciones: “los docentes en general no están respaldados por la autoridad”.

Para finalizar sostienen que un factor de riesgo para el abordaje del acoso entre pares en las escuelas lo constituye la falta de respaldo de la autoridad a la tarea del docente en el aula y fuera de ella.

La percepción de los docentes sobre los adolescentes.

Los docentes tienen una visión negativa de los adolescentes, considerándolos apáticos y sin aspiraciones.

Sin embargo, reconocen la necesidad de cambiar esta perspectiva y destacan el compañerismo y la solidaridad que pueden mostrar.

La influencia de la tecnología en la vida de los adolescentes es resaltada, ya que valoran la dimensión de ser vistos y reconocidos en redes sociales.

Los docentes perciben una modalidad de relación agresiva entre los adolescentes, lo cual dificulta la propuesta de actividades. También observan dificultades en la integración y solidaridad entre ellos. En cuanto al protagonismo juvenil, algunos docentes reconocen la importancia de involucrar a los alumnos en acciones colectivas, mientras que otros perciben la necesidad de escuchar y confiar en ellos. Destacan el potencial de acción de los adolescentes, como la recolección de ropa para un comedor cercano, pero enfatizan que este potencial requiere el apoyo de los adultos y la institución en general.

Síntesis

Intervenciones poner en practica y prevenir el suicidio.

INTERVENCIONES	POLÍTICAS PÚBLICAS	CUIDADOS DE LA SALUD
<ul style="list-style-type: none">■ Restricción del acceso a medios letales.■ Talleres de sensibilización en escuelas.■ Entrenamiento de líderes adultos y juveniles.■ Intervenciones basadas en internet.■ Líneas de ayuda telefónicas.■ Capacitación de operadores en actividades de posvención.■ Campañas en medios.	<ul style="list-style-type: none">■ Creación de redes intersectoriales.	<ul style="list-style-type: none">■ Capacitación de profesionales de la salud.■ Incorporación de profesionales de salud mental en guardias y servicios de adolescencia.■ Aprovechar las oportunidades de consultas de adolescentes en servicios de salud para crear confianza y establecer vínculos.■ Creación de protocolos y difusión de los existentes.

Conclusiones finales.

El suicidio adolescente se ha convertido en la segunda causa de muerte entre los jóvenes de 15 a 19 años, después de los accidentes de tráfico.

En las últimas décadas, se ha observado un aumento en las tasas de suicidio, este fenómeno afecta principalmente a los varones y adolescentes con menor nivel educativo.

No solo las ciudades más grandes como BsAs, sino también las pequeñas localidades presentan casos acumulativos de suicidio adolescente. Aunque las poblaciones más grandes muestran un mayor número de suicidios en términos absolutos, en proporción, hay departamentos con poca población donde el suicidio adolescente es una realidad cercana, en varias recopilaciones de información se explica que debido a las dimensiones geográficas es difícil acceder o la gente no está abierta a responder.

Los factores más importantes relacionados con conducta suicida en adolescentes incluyen son:

Falta de apoyo emocional.

Dificultades en la transición de la adolescencia a la adultez.

Logros educativos, laborales y afectivos.

Además, los trastornos mentales no tratados también desempeñan un papel importante.

Aprendimos que las tipologías de suicidios consumados y tentativas de suicidio ayudan a identificar situaciones de riesgo que requieren intervención preventiva.

Sin embargo, es importante recordar que no representan el total poblacional y es necesario abordar el suicidio adolescente desde una perspectiva multidimensional,

ya que la predicción precisa no es posible a pesar de la identificación de factores de riesgo específicos por la naturaleza compleja que conlleva.

Para prevenir el suicidio adolescente, es necesaria una articulación entre las instituciones y una capacitación continua de los recursos humanos en diferentes ámbitos, como la educación, la salud, la protección y las fuerzas de seguridad. La formación de líderes juveniles como "preventores" y el apoyo de adultos capacitados han demostrado ser estrategias efectivas en diversas comunidades y en estudios prácticos en algunos países, incluido en Argentina.

Existen falencias en las instituciones en términos de capacitación y recursos disponibles para la prevención y asistencia adecuada de los intentos de suicidio. Además, la falta de redes institucionales y la falta de conciencia epidemiológica dificultan la recopilación de datos precisos sobre el suicidio adolescente.

Se requiere un continuo desarrollo de políticas públicas y la mejora de las instituciones, fortalecer la prevención, la capacitación de los actores involucrados y mejorar la recolección de datos para comprender mejor el fenómeno y tomar medidas adecuadas.

Cierre.

Estadísticas sobre el impacto del bullying: Intenté recopilar la mayor cantidad, debido a la naturaleza de los datos es muy difícil contar con datos detallados, creo que la información recogida demuestra la prevalencia y los efectos negativos del bullying en la sociedad. Por ejemplo, cifras sobre el aumento de casos de bullying en los últimos años, el impacto psicológico y emocional en las víctimas, y la necesidad de abordar este problema de manera más efectiva.

Investigaciones previas: Nos enfocamos en datos obtenidos de los principales periódicos nacionales, UNICEF, OMS, Naciones Unidas, Gobierno Argentino, Bullying sin fronteras.

Herramientas existentes: Si bien existen protocolos actualmente que involucran a todas las partes, al ser consultados, todos coinciden en que hace falta involucrar más recursos humanos para satisfacer la demanda.

Experiencias y testimonios de docentes: Intente colocar una opinión que se repite a lo largo de la investigación, en este caso pertenece a un paper que analiza un grupo de 51 docentes.

Casos de éxito: Uno de los casos experimentales que mejores resultados está demostrando actualmente es "*Estrategia de intervención para la prevención del suicidio en adolescentes: la escuela como contexto*. Piedrahita S., Paz, C. y Romero, A. (2012): *Hacia la promoción de la Salud [online]*. Vol.17, N.2, pp.136-148."

El cual dejó buenos resultados en Cali, Colombia y Bs As, Argentina.

En resumen:

El objetivo fue fomentar conductas protectoras a partir de los factores de riesgo para intentos de suicidio identificados en adolescentes y preadolescentes (9 a 14 años), promover su conocimiento y dotar a educadores y padres de familia de herramientas para el trabajo inicial con dichos factores.

Se aplicó con estudiantes, matriculados en una institución educativa de la ciudad de Cali, Colombia, en los primeros años de la escuela secundaria y con adultos

educadores y padres. La intervención educativa posibilitó la identificación de los factores de riesgo en adolescentes y mostró una significativa efectividad al aumentar el nivel de conocimientos de los padres y educadores.

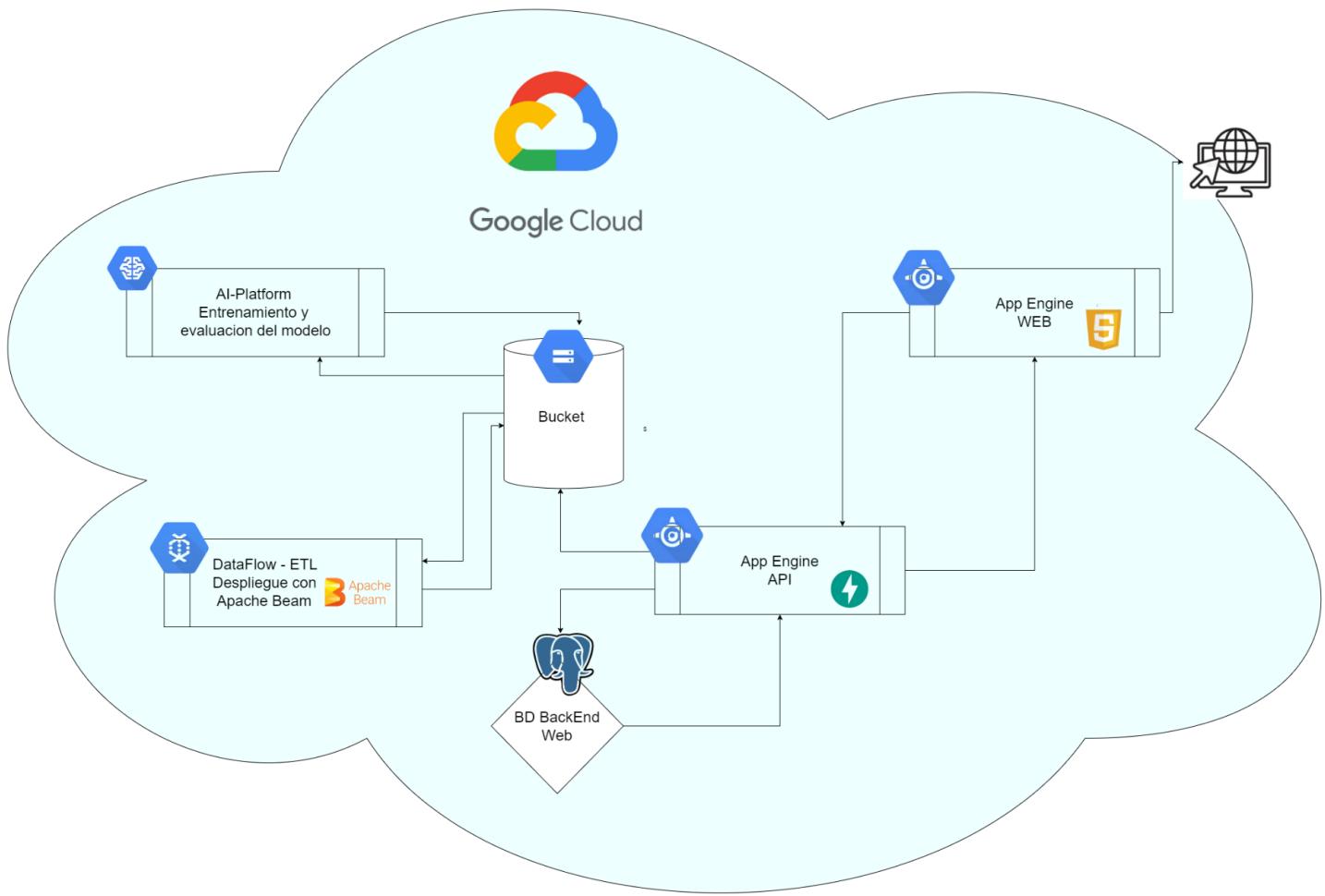
Una mayor información posibilita la identificación y tratamiento oportuno, lo cual lleva a la disminución del evento en este grupo poblacional.

Los hallazgos evidenciaron el desconocimiento de los adultos respecto al suicidio.

Después de la intervención educativa se evidenció más precisión respecto a los conceptos básicos sobre suicidio y de las intervenciones a realizar con adolescentes en riesgo. Lecciones aprendidas: Es importante realizar intervenciones integrales de prevención de la conducta suicida en las escuelas, incluyendo a alumnos, docentes y familiares. Asimismo, es importante desarrollar la evaluación de las experiencias realizadas. Aplicabilidad: Se resalta el papel de la escuela como contexto apropiado para la realización de las intervenciones preventivas de la conducta suicida tanto durante la preadolescencia como en los primeros años de la adolescencia.

3. Arquitectura

3.1. Diseño



3.1.1.1. Premisas para el desarrollo de la arquitectura

- Escalabilidad
- Rendimiento
- Disponibilidad
- Seguridad

Desde un principio tuvimos claro que la arquitectura del proyecto era uno de los aspectos más importantes a tener en cuenta para su buen desarrollo. En la fase inicial del proyecto, tras realizar exploración exhaustiva de los datos y plantear el plan de negocio optamos por que el proyecto se desarrollaría en Google Cloud Platform (GCP), debido a que proporciona numerosas ventajas para su escalabilidad, rendimiento y disponibilidad.

En cuanto al almacenamiento,

Hemos optado por un lado, por una base de datos SQL de postgre, ya que cumple con las premisas de la arquitectura (escalabilidad, rendimiento y disponibilidad) y además , al estar alojada en GCP, permite acceder a características avanzadas de la plataforma, como la replicación de bases de datos y la integración con otros servicios de GCP.

Por otro lado, para el almacenamiento de cualquier otro tipo de archivos, como pueden ser el modelo o el escaler, hemos optado por un bucket. Este servicio es altamente escalable, lo que significa que pueden manejar grandes volúmenes de datos y ajustar automáticamente la cantidad de recursos necesarios para hacerlo. Además, los buckets en GCP ofrecen características avanzadas, como la integración con otros servicios de la plataforma, como Dataflow, BigQuery y Cloud Storage, lo que permite una mayor flexibilidad en la gestión de los datos.

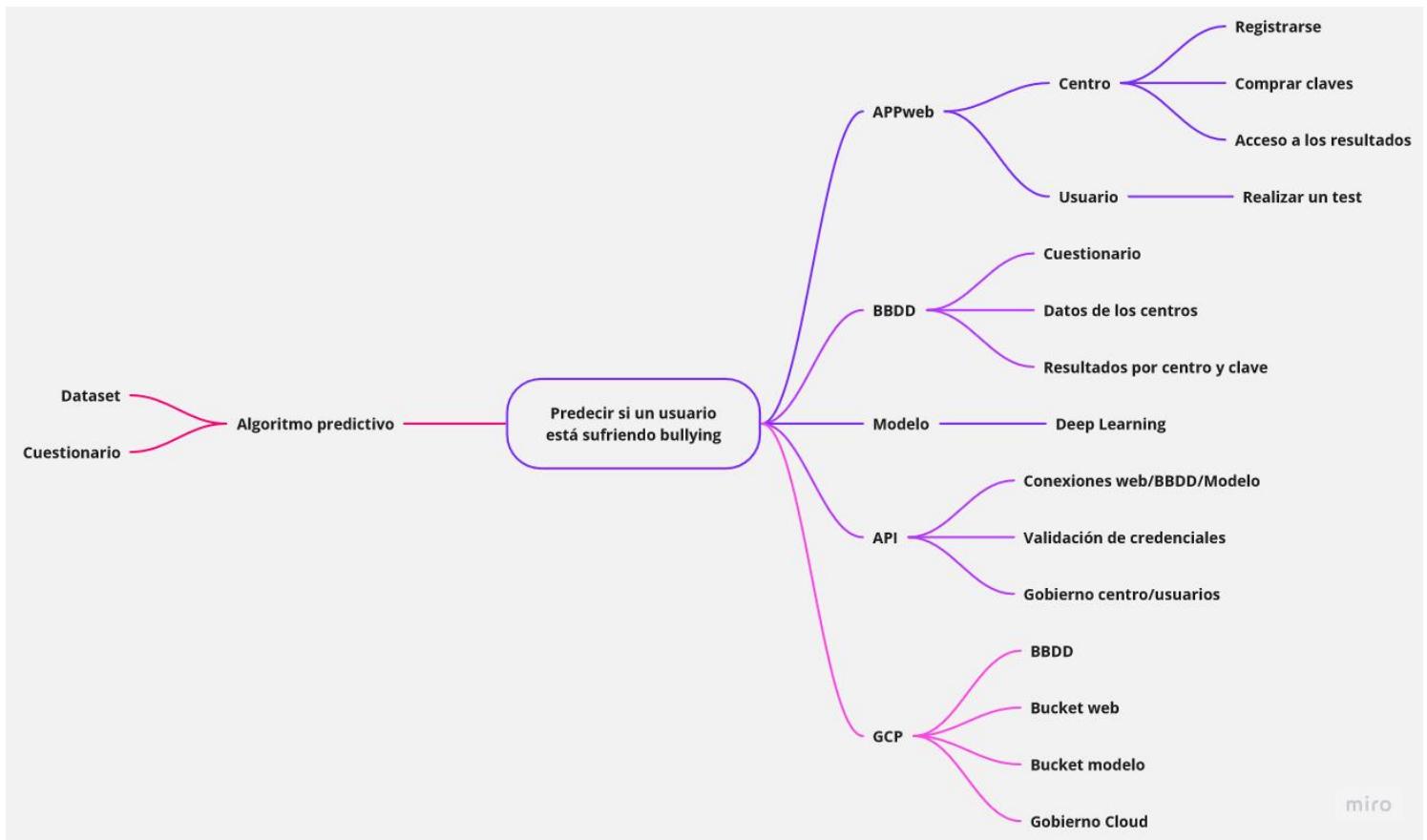
En cuanto a la API, hemos optado por el framework FastAPi ya que proporciona una alta velocidad y escalabilidad, además de una documentación clara y fácil de seguir. Para desplegar la API en cloud hemos optado por App Engine, en el que se obtiene un alto nivel de disponibilidad y escalabilidad, ya que, igual que muchos productos de GCP, App Engine se encarga de escalar automáticamente los recursos según la demanda del servicio.

En cuanto a la creación de la web, tuvimos problemas al desplegarla en App Engine. Dada la escasez de tiempo, priorizamos la consecución de una interfaz amigable, dinámica y segura frente al despliegue de la misma en el entorno cloud. A pesar de ello hemos construido la web usando HTML, CSS y JS. Este último nos ha brindado las herramientas necesarias para ejecutar peticiones asíncronas y procesar las respuestas de la API, así como para acometer la tarea de construir un frontend absolutamente dinámico que va construyéndose en el DOM en función de las necesidades del usuario y evitando su modificación en caso de adición de nuevos elementos. Un ejemplo claro de ello es la generación dinámica del cuestionario que los alumnos deben realizar para evaluar el riesgo que éstos tienen de sufrir bullying; independientemente de las preguntas y las posibles respuestas que se reciben de la API (mediante un json) el despliegue lógico y visual se mantiene.

Este diseño nos permitirá minimizar exponencialmente los tiempos de desarrollo en caso de añadir diferentes cuestionarios y es un ejemplo paradigmático del enfoque arquitectónico del proyecto, en el que cada paso se da pensando en dejar el terreno llano para próximas mejoras y/o adiciones de funcionalidad.

Finalmente, para la puesta en producción del modelo, aunque a priori, tanto por la naturaleza de los datos como por la cantidad, no era necesario generar un pipeline paralelizado, igualmente hemos optado por utilizar Apache Beam, ya que nuestra idea de negocio implicaría realizar reentrenamientos periódicos del modelo con los datos que se vayan recabando con los test realizados en la web. Para el despliegue en cloud usaremos Dataflow y AI-Platform, ya que permite escalar el modelo automáticamente según la demanda del servicio y proporciona características avanzadas, como la integración con otros servicios de GCP, como BigQuery y Cloud Storage

3.2. Desarrollo



3.2.1. API

3.2.1.1. Introducción

El objetivo principal de esta API es proporcionar a los colegios una herramienta para prevenir y detectar el bullying en sus alumnos. La API permitirá a los colegios registrarse en la plataforma web y solicitar identificadores únicos para sus alumnos, lo que les permitirá responder a una encuesta. La API procesa los resultados de la encuesta a través de un modelo de Machine Learning en GCP para determinar si el alumno sufre o no de bullying. Todo el proyecto será desplegado en GCP.

3.2.1.2. Estructura del proyecto

El proyecto tiene la siguiente estructura:

...

```
fastapi/
    ├── __init__.py
    ├── main.py
    ├── .env
    └── requirements.txt
Project/
    ├── config.py
    ├── db.py
    ├── utils.py
    ├── models/
    │   ├── __init__.py
    │   ├── school_model.py
    │   ├── student_model.py
    │   ├── survey_model.py
    │   ├── answer_model.py
    │   ├── master_model.py
    │   └── satellite_tables_model.py
    ├── controllers/
    │   ├── __init__.py
    │   ├── auth_school.py
    │   ├── register_school.py
    │   ├── school_controllers.py
    │   └── survey_controllers.py
    ├── schemas/
    │   ├── __init__.py
    │   ├── school_schema.py
    │   ├── survey_schema.py
    │   ├── application_id_schema.py
    │   └── token_schema.py
    └── endpoints/
        ├── __init__.py
        └── school_endpoints.py
```

```

    └── surveys_endpoints.py
    └── machine_learning/
        ├── __init__.py
        └── model.py
...

```

- **__init__.py**: es un archivo que Python utiliza para indicar que la carpeta es un paquete.
- **config.py**: archivo que contiene la configuración de la aplicación, como la cadena de conexión a la base de datos.
- **main.py**: archivo principal que ejecuta la aplicación.
- **db.py**: archivo que define y establece la conexión con la base de datos.
- **.env**: archivo que contiene variables de entorno que se cargan en la aplicación en tiempo de ejecución.
- **utils.py**: archivo que contiene funciones de utilidad genéricas.
- **requirements.txt**: archivo que contiene las dependencias del proyecto.
- **models/**: carpeta que contiene los modelos de la base de datos.
- **controllers/**: carpeta que contiene los controladores de la aplicación.
- **schemas/**: carpeta que contiene los esquemas de validación de datos.
- **endpoints/**: carpeta que contiene las rutas y controladores para los endpoints de la aplicación.
- **machine_learning/**: carpeta que contiene los archivos relacionados con el modelo de aprendizaje automático.

3.2.1.3. Modelos de datos:

Para realizar el modelado de los datos usaremos la librería peewee. Peewee es un ORM (Object-Relational Mapping) para Python que se utiliza para trabajar con bases de datos relacionales. Un ORM es una técnica de programación que permite representar objetos de una aplicación en una base de datos, y viceversa, sin tener que escribir código SQL directamente. En lugar de eso, el ORM se encarga de traducir las operaciones de bases de datos en código Python.

Peewee es una biblioteca ligera y fácil de usar que admite múltiples bases de datos relacionales, como SQLite, MySQL y PostgreSQL, entre otras. Se integra bien con diferentes marcos web.

Entre las características notables de Peewee se incluyen:

- Soporte para múltiples bases de datos y tipos de campos, incluidos campos personalizados.
- Fácil creación y modificación de esquemas de bases de datos.

- Selección, actualización, inserción y eliminación de datos.
- Soporte para transacciones y bloqueo de bases de datos.
- Una sintaxis clara y concisa para consultas, y una API simple y fácil de usar.

Hemos definido los siguientes modelos:

School

- **`school_id`**: campo autoincremental que actúa como clave primaria de la tabla.
- **`desc_school`**: campo de tipo CharField que almacena la descripción de la escuela.
- **`cif`**: campo de tipo CharField que almacena el código de identificación fiscal de la escuela.
- **`phone`**: campo de tipo CharField que almacena el número de teléfono de la escuela.
- **`zip_code`**: campo de tipo CharField que almacena el código postal de la escuela.
- **`email`**: campo de tipo CharField que almacena la dirección de correo electrónico de la escuela.
- **`country_id`**: campo de tipo ForeignKeyField que hace referencia a la tabla de países y almacena el ID del país donde se encuentra la escuela.
- **`city`**: campo de tipo CharField que almacena la ciudad donde se encuentra la escuela.
- **`password`**: campo de tipo CharField que almacena la contraseña de acceso a la plataforma para la escuela.
- **`credits`**: campo de tipo IntegerField que almacena el número de créditos disponibles para la escuela.
- **`type_id`**: campo de tipo ForeignKeyField que hace referencia a la tabla de tipos de escuela y almacena el ID del tipo de escuela.
- **`dt_insert`**: campo de tipo DateTimeField que almacena la fecha y hora de inserción del registro.
- **`dt_update`**: campo de tipo DateTimeField que almacena la fecha y hora de la última actualización del registro.
- **`comments`**: campo de tipo TextField que almacena comentarios adicionales sobre la escuela.
- **`disable`**: campo de tipo BooleanField que indica si la escuela está deshabilitada o no.

Master

- **`entry_id`**: campo autoincremental que actúa como clave primaria de la tabla.
- **`school_id`**: campo de tipo ForeignKeyField que hace referencia a la tabla School y almacena el ID de la escuela asociada al registro.
- **`student_id`**: campo de tipo ForeignKeyField que hace referencia a la tabla Student y almacena el ID del estudiante asociado al registro.

- **model_id**: campo de tipo ForeignKeyField que hace referencia a la tabla SurveyModels y almacena el ID del modelo de encuesta utilizado para el registro.
- **prediction**: campo de tipo IntegerField que almacena el resultado de la predicción realizada para el registro.
- **prob_prediction**: campo de tipo IntegerField que almacena la probabilidad asociada al resultado de la predicción.
- **date_insert**: campo de tipo DateTimeField que almacena la fecha y hora de inserción del registro.
- **date_update**: campo de tipo DateTimeField que almacena la fecha y hora de la última actualización del registro.
- **comments**: campo de tipo TextField que almacena comentarios adicionales sobre el registro.

School_types

El modelo "School_types" define una tabla en la base de datos que almacena información sobre los diferentes tipos de escuelas que pueden ser registradas en el sistema. Esta tabla tiene los siguientes campos:

- **type_id**: campo autoincremental que actúa como clave primaria de la tabla.
- **desc_type**: campo de tipo CharField que almacena la descripción del tipo de escuela.
- **dt_insert**: campo de tipo DateTimeField que almacena la fecha y hora de inserción del registro en la tabla.
- **dt_update**: campo de tipo DateTimeField que almacena la fecha y hora de la última actualización del registro en la tabla.

Countries

El modelo "Countries" define una tabla en la base de datos que almacena información sobre los países donde se encuentran las diferentes escuelas registradas en el sistema. Esta tabla tiene los siguientes campos:

- country_id**: campo autoincremental que actúa como clave primaria de la tabla.
- desc_country**: campo de tipo CharField que almacena la descripción del país.
- dt_insert**: campo de tipo DateTimeField que almacena la fecha y hora de inserción del registro en la tabla.
- dt_update**: campo de tipo DateTimeField que almacena la fecha y hora de la última actualización del registro en la tabla.

Student

El modelo "Student" define una tabla en la base de datos que almacena información sobre los estudiantes registrados en una escuela y sus comentarios y calificaciones sobre las encuestas realizadas. Esta tabla tiene los siguientes campos:

- **student_id**: campo de tipo TextField que actúa como clave primaria de la tabla y almacena el ID del estudiante.
- **school_id**: campo de tipo ForeignKeyField que hace referencia a la tabla School y almacena el ID de la escuela asociada al estudiante.

- **comments**: campo de tipo TextField que almacena los comentarios del estudiante sobre las encuestas realizadas.
- **dt_insert**: campo de tipo DateTimeField que almacena la fecha y hora de creación del registro.
- **dt_update**: campo de tipo DateTimeField que almacena la fecha y hora de la última actualización del registro.
- **credits**: campo de tipo IntegerField que almacena los créditos del estudiante.
- **times_done**: campo de tipo IntegerField que almacena la cantidad de veces que el estudiante ha completado las encuestas.

SurveyModels

El modelo "SurveyModels" define una tabla en la base de datos que almacena información relacionada con los modelos de encuestas. Esta tabla tiene los siguientes campos:

- **model_id**: campo autoincremental que actúa como clave primaria de la tabla.
- **desc_survey**: campo de tipo CharField que almacena la descripción del modelo de encuesta.
- **date_insert**: campo de tipo DateTimeField que almacena la fecha de inserción del registro.
- **date_update**: campo de tipo DateTimeField que almacena la fecha de actualización del registro.
- **comments**: campo de tipo TextField que almacena comentarios adicionales sobre el modelo de encuesta.

Survey_Questions

El modelo "Survey_Questions" define una tabla en la base de datos que almacena información relacionada con las preguntas de los modelos de encuestas. Esta tabla tiene los siguientes campos:

- **model_id**: campo de tipo ForeignKeyField que hace referencia a la tabla SurveyModels y almacena el ID del modelo de encuesta asociado a la pregunta.
- **answer_id**: campo que hace referencia al tipo de respuestas de la pregunta del test.
- **order_num**: campo de tipo IntegerField que almacena el orden de la pregunta dentro del modelo de encuesta.
- **quest**: campo de tipo TextField que almacena el texto de la pregunta.

Response_answers

El modelo "Response_answers", almacena las posibles respuestas a las preguntas del test, contiene los siguientes campos:

- **answer_id**: campo de tipo clave primaria
- **answer**: columna donde se almacena los distintos tipos de respuestas

answers

El modelo "answers" almacena las respuestas de los alumnos, contiene los siguientes campos:

- **entry_id**: clave primaria, y a su vez clave foranea de la tabla master
- **model_id**: clave foranea de la tabla survey_models

- **answer_json**: columna donde se almacena el json con las respuesta del alumno.

3.2.1.4. Autentificación

El endpoint para la autenticación es /token. La autenticación utiliza el esquema de autenticación OAuth2, es decir, permite a las aplicaciones obtener tokens de acceso en nombre de un usuario al enviar las credenciales del usuario directamente al servidor de autorización.

En este flujo, el usuario proporciona sus credenciales de inicio de sesión (cif y contraseña) a la app, que a su vez envía una solicitud de token de acceso al servidor de autorización. El servidor de autorización autentica al usuario y, si las credenciales son correctas, emite un token de acceso al cliente. Este token puede usarse para acceder a los recursos protegidos por el servidor de recursos (API).

Para obtener un token de acceso, se debe enviar una solicitud POST a /token con los siguientes datos de formulario:

- cif: identificación fiscal del centro.
- password: Contraseña.

Si las credenciales son válidas, el servidor responderá con un token de acceso válido.

Los endpoints que requieren autenticación utilizan el esquema de autenticación Bearer con el token de acceso recibido en el paso anterior.

Los endpoints que requieren autenticación son:

/payment
/survey/applicate_id.
/survey/profile

3.2.1.5. Endpoints

Para la creación de los esquemas de datos de las solicitudes y respuestas, hemos utilizado la librería pydantic. Pydantic es una biblioteca de Python que se utiliza para validar y serializar datos diseñada para trabajar con estructuras de datos que se utilizan comúnmente en aplicaciones web, como JSON y YAML. Con estos esquemas de datos conseguimos que sea más fácil desarrollar aplicaciones web que sean seguras y estén bien documentadas.

La aplicación consta de los siguiente endpoints:

School_endpoints.py

Estos endpoints forman parte de una API de gestión de colegios y autenticación de usuarios mediante tokens:

- Método: POST, Endpoint: /register/

Descripción: Crea un nuevo colegio en la app.

Parámetros: Se reciben los campos del colegio en formato JSON.

Respuesta: Retorna la información del colegio registrado.

formato petición - application/json

```
{  
    "desc_school": "Name School",  
    "cif": "B0000000",  
    "phone": "9999999999",  
    "zip_code": "15011",  
    "email": "info@school.com",  
    "password": "strongpass",  
    "country_id": "España",  
    "type_id": "Concertado",  
    "city": "La Coruña"  
}
```

formato respuesta - application/json

```
{  
    "desc_school": "Name School",  
    "cif": "cccccc",  
    "phone": "9999999999",  
    "zip_code": "15011",  
    "email": "info@school.com",  
    "school_id": 20,  
    "country_id": 1,  
    "type_id": 2  
}
```

- Método: GET, Endpoint: /school/me

Descripción: Retorna la información del colegio actualmente autenticado.

Parámetros: Se espera un token de autenticación en la cabecera de la petición.

Respuesta: Retorna la información del colegio actualmente autenticado.

formato petición -application/json

```
{  
    "access_token": "string",  
    "token_type": "string"  
}
```

- Método: POST, Endpoint: /token

Descripción: Inicia sesión y devuelve un token de acceso.

Parámetros: Se espera que se envíen las credenciales del usuario en formato de formulario.

Respuesta: Retorna un objeto Token con el token de acceso y el tipo de token.

formato - application/x-www-form-urlencoded
username y password

formato respuesta -application/json

```
{  
  "access_token": "string",  
  "token_type": "string"  
}
```

- Método: POST, Endpoint: /payment

Descripción: Procesa un pago.

Parámetros: Se espera que se envíe el monto del pago y un token de autenticación en la cabecera de la petición.

- Método: POST, Endpoint: /survey/applicate_id

Descripción: Crea los identificadores para que los estudiantes tengan acceso a la encuesta.

Parámetros: Se espera un objeto ApplicationIDCreate con la cantidad de estudiantes, el curso y la clase a la que pertenecen, y un token de autenticación en la cabecera de la petición.

Respuesta: Retorna un diccionario con los identificadores de los estudiantes creados.

formato petición -application/json

```
{  
  'PRIMARIA': {  
    'Curso 1': {'Aula 1': 3, 'Aula 2': 2},  
    'Curso 2': {'Aula 1': 3, 'Aula 2': 5, 'Aula 3': 4}  
  },  
  'SECUNDARIA': {  
    'Curso 1': {'Aula 1': 5, 'Aula 2': 5, 'Aula 3': 5},  
    'Curso 2': {'Aula 1': 6, 'Aula 2': 6, 'Aula 3': 6, 'Aula 4': 6},  
    'Curso 3': {'Aula 1': 10, 'Aula 2': 10},  
    'Curso 4': {'Aula 1': 5, 'Aula 2': 5, 'Aula 3': 5}  
  },  
  'BACHILLERATO': {  
    'Curso 1': {'Aula 1': 10, 'Aula 2': 10, 'Aula 3': 10},  
    'Curso 2': {'Aula 1': 9, 'Aula 2': 9, 'Aula 3': 9},  
    'Curso 3': {'Aula 1': 4, 'Aula 2': 4, 'Aula 3': 4, 'Aula 4': 4, 'Aula 5': 4}  
  }  
}
```

formato respuesta -application/json

```
{  
  'PRIMARIA': {'Curso 1': {'Aula 1': ['202341PRI1Au1001']}},  
  'SECUNDARIA': {'Curso 1': {'Aula 1': ['202341SEC1Au1001']}},
```

```
'BACHILLERATO': {'Curso 1': {'Aula 1': ['202341BAC1Au1001']}}  
}
```

- Método: GET, Endpoint: /profile

Descripción: Realiza consulta a la base de datos de las fechas en las que el colegio realizo los test para mostrarlos en la web

Parámetros: Solo recibe por parámetro el token de autenticación

Respuesta: Devuelve un json con las fechas de los test solicitados

formato respuesta -application/json

- Método: POST, Endpoint: /query

Descripción: Realiza la consulta de los student_id y sus predicciones.

Parámetros: Recibe una fecha en formato 'April - 2023', y un parámetro llamado order_by que indica a la api como ordenar el resultado de la búsqueda, únicamente admite 'student_id' y 'prediction'

Respuesta: Devuelve un json con los id de estudiante y sus predicciones

formato petición -application/json

...

```
{  
    "order_by": "string",  
    "date": "string"  
}
```

...

formato respuesta -application/json

...

...

Survey_endpoints.py

Los siguientes endpoints están relacionados con las encuestas

- Método: POST, Endpoint: /survey/questions/{student_id}

Descripción: Permite acceder a la encuesta a través del identificador del alumno.

Parámetros: student_id: Identificador del alumno que se desea obtener la encuesta.

Respuesta: Retorna un diccionario con las preguntas de la encuesta y las opciones de respuesta para cada una.

formato petición - string

```
"student_id": "string",
```

formato respuesta -application/json

```
{  
  "questions": [  
    {  
      "quest": "¿Qué edad tienes?",  
      "answer": "[]"  
    },  
    {  
      "quest": "¿En qué grado estás?",  
      "answer": "[]"  
    },  
    {  
      "quest": "¿Cuánto mides sin zapatos? (Nota: los datos están en metros.)?",  
      "answer": "[]"  
    },  
    {  
      "quest": "¿Cuánto pesas sin zapatos? (Nota: los datos están en kilogramos.)?",  
      "answer": "[]"  
    },  
    {  
      "quest": "¿Cuál es tu género?",  
      "answer": "[\"Mujer\", \"Hombre\"]"  
    },  
    ...  
  ]  
}
```

- Método: POST, Endpoint: /survey/{student_id}/submit

Descripción: Recoge json con respuestas de encuesta del alumno, el nombre del colegio y el student_id.

Parámetros: student_id: Identificador del alumno que se desea obtener la encuesta.

ques: Objeto SurveyQuestions con las preguntas de la encuesta.

Respuesta: Retorna un diccionario con las preguntas de la encuesta y las opciones de respuesta para cada una. Se espera que el estudiante responda la encuesta y se guarden las respuestas en la base de datos.

3.2.1.6. Tecnologías

FastApi

Para la realización de esta API, hemos optado por usar el marco web basado en Python, Fastapi. Su estructura de desarrollo es similar a Flask.

Fastapi es fácil de usar y su [documentación](<https://fastapi.tiangolo.com/es/>) es clara. Además ofrece un alto rendimiento, y genera la documentación de forma automática con un esfuerzo mínimo por parte del desarrollador. Esta información se puede encontrar en el directorio /docs de la aplicación. La documentación

contiene información detallada sobre puntos finales de API, códigos de retorno, parámetros de respuesta y otros detalles.

App Engine de Google Cloud Platform

Para realizar el despliegue del proyecto de la API en cloud hemos optado por [App Engine](<https://cloud.google.com/appengine>). Google App Engine es otro de los servicios que conforman la familia de Google Cloud Platform. Este servicio es del tipo Plataforma como Servicio o Platform as a Service (PaaS), nos permite publicar aplicaciones web en línea sin necesidad de preocuparnos por la parte de la infraestructura y con un enfoque 100% en la construcción de nuestra aplicación y en la posibilidad de correrla directamente sobre la infraestructura de Google, es decir, la que Google usa para sus propios productos.

Como cualquier otra Plataforma como Servicio, App Engine nos facilita construir, mantener y escalar nuestra aplicación en la medida que sea necesario. Cuando usamos Google App Engine (GAE) no nos tenemos que preocupar por la escalabilidad de nuestra aplicación ya que cuenta con un balanceador de carga y escalamiento automático.

Así nuestra aplicación solamente será atendida por las máquinas necesarias para tener un perfecto comportamiento y para que la respuesta de nuestra aplicación sea la más óptima.

3.2.1.7. Dependencias

Las dependencias utilizadas en el proyecto son:

- **fastapi==0.95.0**: Es un framework web para construir APIs rápidas y escalables con Python 3.6+ basado en estándares abiertos. Proporciona herramientas para la validación de datos, la documentación de API y la autenticación de usuario.
- **uvicorn==0.21.1**: Es un servidor web asíncrono basado en ASGI (Asynchronous Server Gateway Interface) que permite servir aplicaciones web construidas con el framework FastAPI.
- **google-auth**: Es una biblioteca de autenticación para Python que permite autenticar con la API de Google Cloud Platform y otras APIs de Google.
- **google-auth-oauthlib**: Es una biblioteca de autenticación de OAuth 2.0 para Google APIs.
- **google-auth-httplib2**: Es una biblioteca de autenticación de HTTP para Google APIs.
- **google-cloud-storage**: Es una biblioteca que permite interactuar con Google Cloud Storage desde Python.
- **pandas**: Es una biblioteca de análisis de datos de código abierto para Python que proporciona estructuras de datos y herramientas para el análisis de datos.

- **numpy**: Es una biblioteca de cálculo numérico para Python que proporciona una gran cantidad de funciones matemáticas y de álgebra lineal.
- **scikit-learn**: Es una biblioteca de aprendizaje automático de código abierto para Python que proporciona herramientas para la minería de datos y el análisis de datos.
- **psycopg2-binary==2.9.5**: Es un adaptador de base de datos PostgreSQL para Python que permite interactuar con bases de datos PostgreSQL desde Python.
- **dotenv**: Es una biblioteca que permite cargar variables de entorno desde un archivo .env en la raíz del proyecto.
- **jwt==1.3.1**: Es una biblioteca que permite codificar y decodificar tokens de autenticación JSON Web Tokens (JWT) en Python.
- **peewee==3.16.0**: Es una biblioteca de ORM (Object Relational Mapper) de Python que proporciona una forma sencilla de interactuar con bases de datos relacionales desde Python.
- **pydantic==1.10.7**: Es una biblioteca que proporciona herramientas para la validación de datos y la serialización de objetos en Python.
- **bcrypt==1.7.4**: Es una biblioteca de hash de contraseñas en Python que proporciona herramientas para la generación y verificación de contraseñas seguras.
- **python-jose==3.3.0**: Es una biblioteca de Python para JSON Object Signing and Encryption (JOSE) que proporciona herramientas para codificar y decodificar tokens de autenticación JSON Web Tokens (JWT) y para cifrar y descifrar datos en JSON.

3.2.2. Web

3.2.2.1. Introducción

Esta será la aplicación web que dará forma al proyecto. Su objetivo es proporcionar a las instituciones educativas una herramienta para detectar de forma más rápida si alguno de sus alumnos está sufriendo acoso escolar. Esta aplicación web trabaja en conjunto con la API para que los centros puedan registrarse y tener un entorno privado donde solicitar claves de acceso y visualizar resultados. Y a su vez, dónde cada alumno de forma individual y en cualquier lugar, sin necesidad de estar logueado pueda realizar el test mediante la clave que le proporciona el centro. De esta forma los resultados del test son totalmente privados y solo el centro tiene acceso a ellos, ya que es solo la institución quién sabe a qué alumno entrega cada clave.

3.2.2.2. Estructura del proyecto

El proyecto tiene la siguiente estructura:

```
...
WEB/
    index.html
    arquitecturaWeb.drawio
    readme.md
    assets/
        centro.html
        generateKeys.html
        home.html
        login.html
        realizartest.html
        Resultados.html
        shop.html
        successful.html
        successKeys.html
        test.html
    css/
        images/
            fondodos.webp
            logo.png
        centro.css
        generateKeys.css
        home.css
        login.css
        realizartest.css
        Resultados.css
        shop.css
        styles.css
        successful.css
        successKeys.css
        test.css
    js/
        index.js
        apiConnection.js
        validations.js
...

```

****index.html**:** es la página principal de la web

****arquitecturaWeb.drawio**:** es el esquema de la web

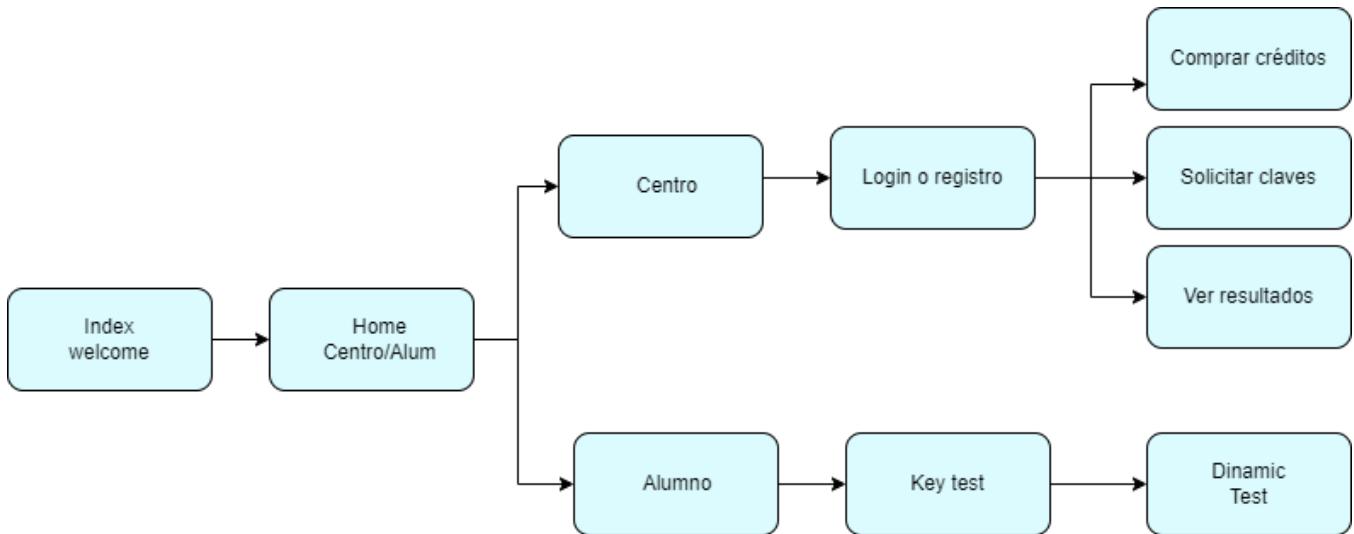
****readme.md**:** es la leyenda de toda la aplicación web

****assets/**:** aquí se encuentran el resto de págs que componen la web

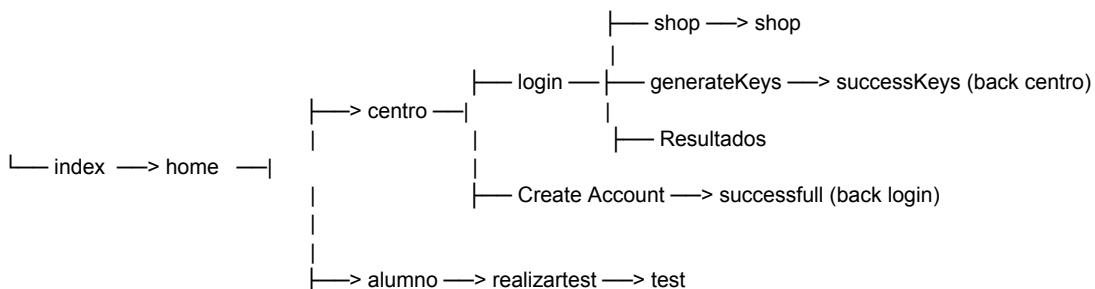
****css/**:** contiene todos los archivos css que dan estilo a la web

3.2.2.3. Arquitectura

Esta sería la arquitectura de la web



Este sería el esquema de la web con redirecciones:



3.2.2.4. Tecnologías

Desarrollar una aplicación web implica la creación de una interfaz de usuario que sea atractiva y fácil de usar, y que permita a los usuarios interactuar con la aplicación de manera efectiva. Para lograr esto, se han utilizado HTML, CSS y JavaScript.

HTML es el lenguaje de marcado estándar utilizado para crear páginas web. En este caso, se ha utilizado la última versión HTML5.

CSS es un lenguaje de hojas de estilo utilizado para diseñar la apariencia de la página web. En este caso, se ha utilizado CSS3, que ofrece una amplia gama de funcionalidades, como la capacidad "flex" de crear diseños responsivos y adaptables.

JavaScript es un lenguaje de programación utilizado para crear interactividad y dinamismo en una página web. En este caso, todos los scripts que dan el dinamismo a la página están desarrollados en JavaScript puro. Estos scripts se han utilizado para crear efectos visuales, como animaciones de texto, validar formularios, enviar solicitudes a la api y otras funcionalidades.

Como ya se ha adelantado en la sección de arquitectura, la web se ha diseñado con el máximo dinamismo. Queríamos que nuestros clientes tengan una experiencia de usuario rápida, sencilla y amigable, por lo que la asincronía era innegociable. Al mismo tiempo, en todo momento hemos tenido la intención de ampliar el contenido que ofrecemos (diferentes cuestionarios para los alumnos, métricas específicas y descriptivas de cada centro, curso y aula y un largo etcétera). Por eso la arquitectura WEB ha tenido en cuenta estas aspiraciones y se ha basado en una construcción del DOM generada en base a los datos que se reciben de la API. Convirtiéndose esta en el cerebro de todas las conexiones e interacciones y propiciando que el frontend sea sumamente adaptativo a las modificaciones en el backend sin necesidad de modificarlo.

Otro punto a destacar es la seguridad. Dada la naturaleza sensible de los datos, se ha implementado un sistema de autenticación 'tokenOAuth' almacenado en la sesión del navegador. Dicho token se obtiene y envía en todas las peticiones a la API que gestionan la información privada asegurando que cada centro y solo cada centro tiene acceso a visualizar y descargar su información privada.

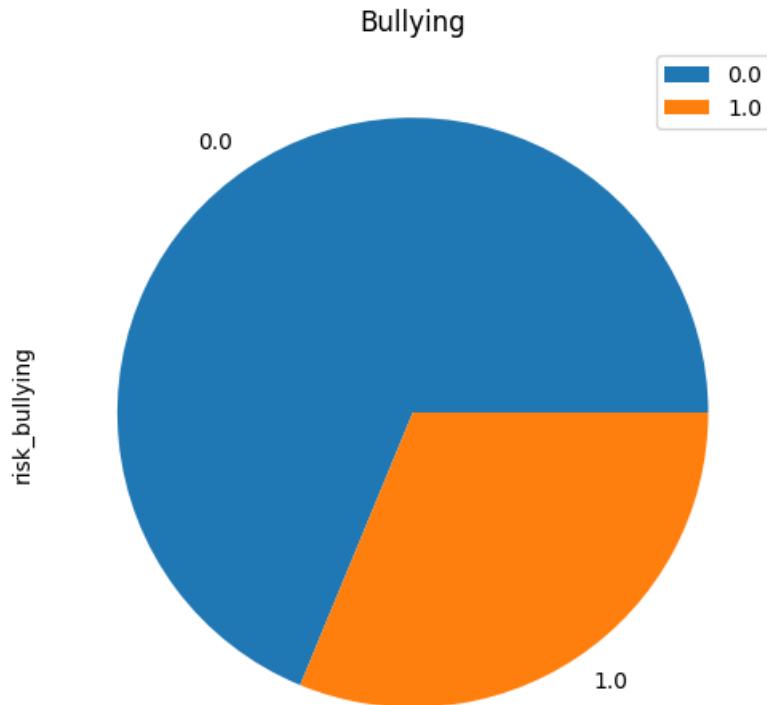
3.2.3. Modelo

El modelo final es una red neuronal diseñada para abordar un problema de clasificación binaria mediante el uso de un conjunto de datos desbalanceado. Toda la información correspondiente al modelo se encuentra detallada en la sección específica de la memoria del proyecto. Como se mencionó previamente, se inició el trabajo con un conjunto de datos diferente, y se generaron varios modelos, los modelos desarrollados utilizando este conjunto se encuentran en el archivo "bullying_1". No obstante, posteriormente se descubrió otro conjunto de datos más amplio. Por lo que se crearon varios modelos adicionales que se describen en la sección correspondiente del modelo, y que se encuentran almacenados en los archivos "bullying_2_ConBalanceo", "bullying_2_KNN" y "Modelo_final". En particular, los modelos generados con la técnica de creación de funciones para balancear las variables están en el archivo "bullying_2_ConBalanceo", mientras que los modelos creados mediante la técnica KNNImputer están en el archivo "bullying_2_KNN". Por último, el modelo final seleccionado se encuentra en el archivo "Modelo_final". Todo esto se detalla en la sección del modelo de este documento.

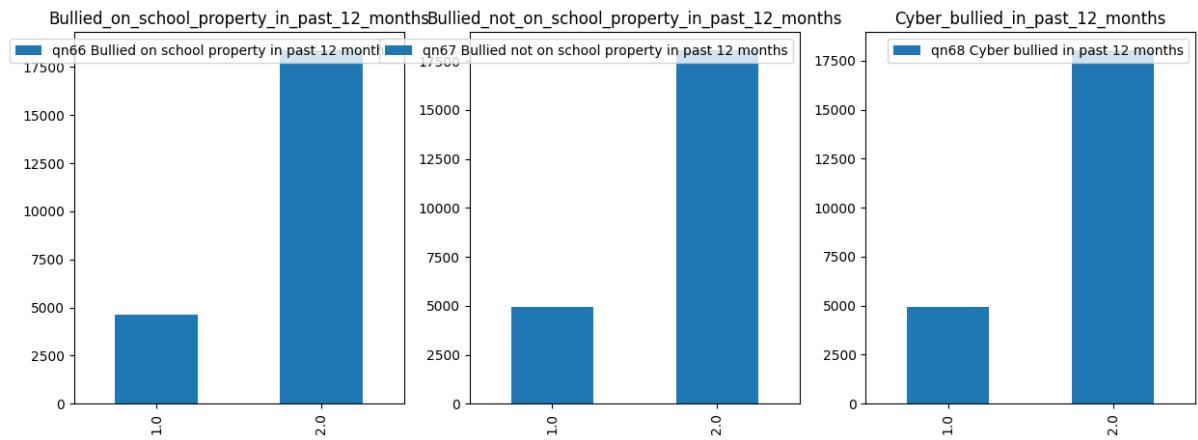
4. Visualización BI

4.1. KPIs

Alumnos con bullying:



Dividimos los tipos de bullying:



Número de en Bullied_on_school_property_in_past_12_months: 18388

Número de casos en Bullied_on_school_property_in_past_12_months: 4613
porcentaje 20.06%

Número de en Bullied_not_on_school_property_in_past_12_months: 18065

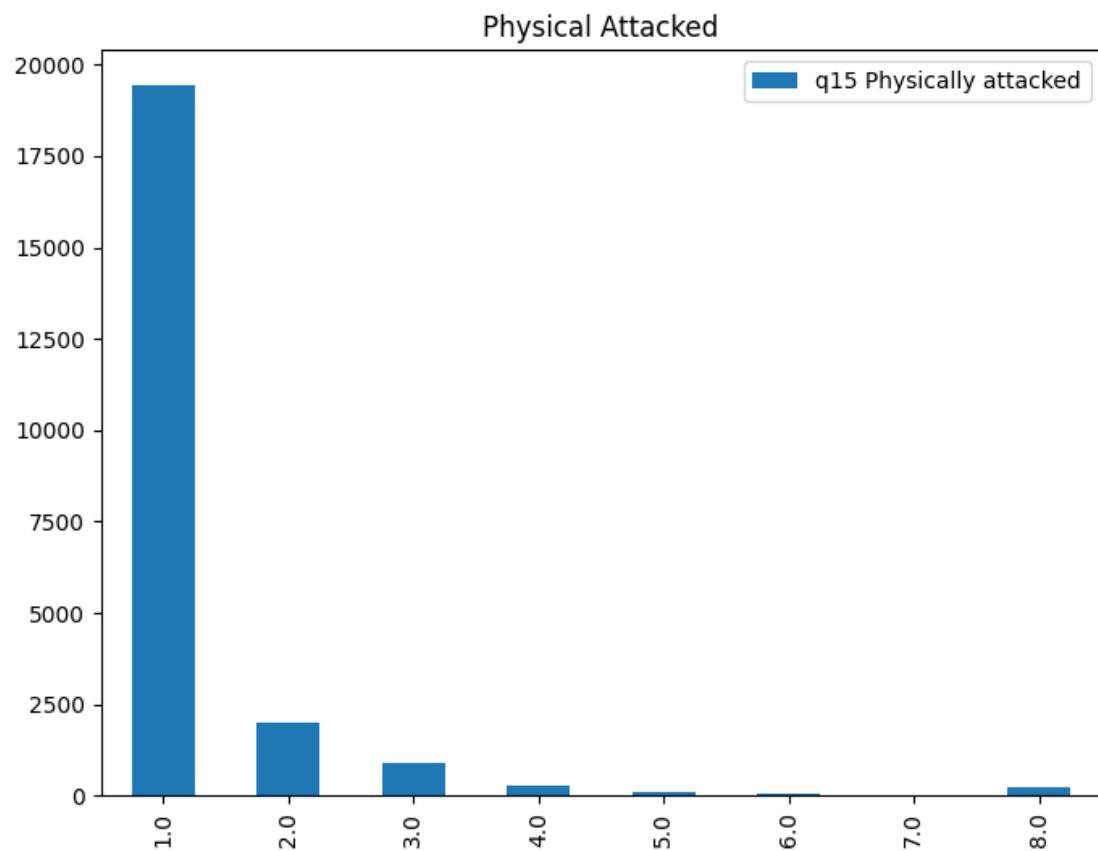
Número de casos en Bullied_not_on_school_property_in_past_12_months: 4936
porcentaje 21.46%

Número de en Cyber_bullied_in_past_12_months: 18046

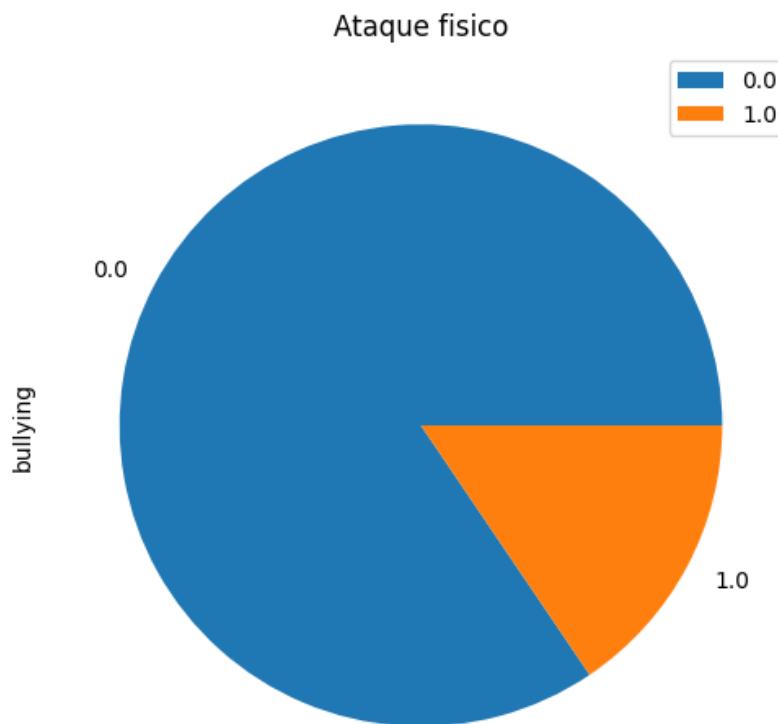
Número de casos en Cyber_bullied_in_past_12_months: 4955

porcentaje 21.54%

Alumnos que presentan ataques físicos, grado grave de bullying:

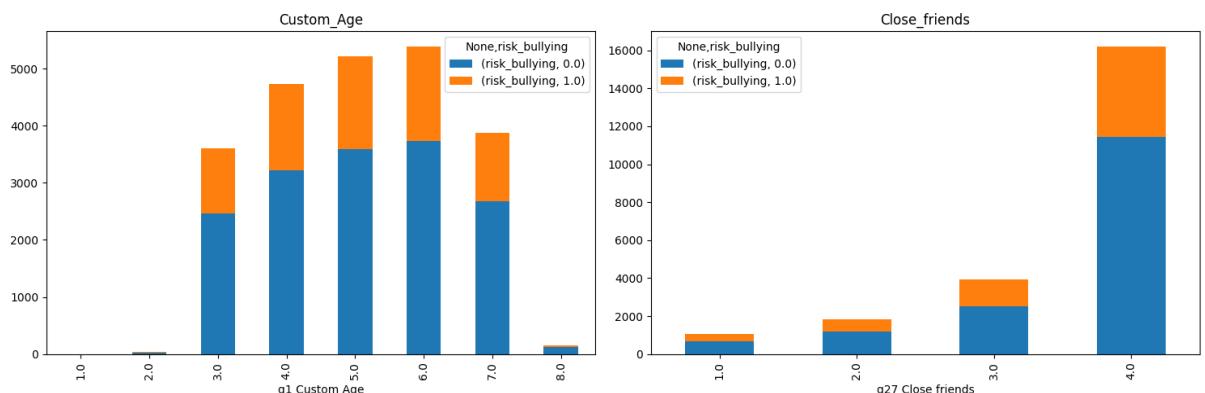


Realizamos una gráfica unificando los ataques físicos, es decir si no hay ningún ataque físico es 0 y si presentan algún ataque 1:

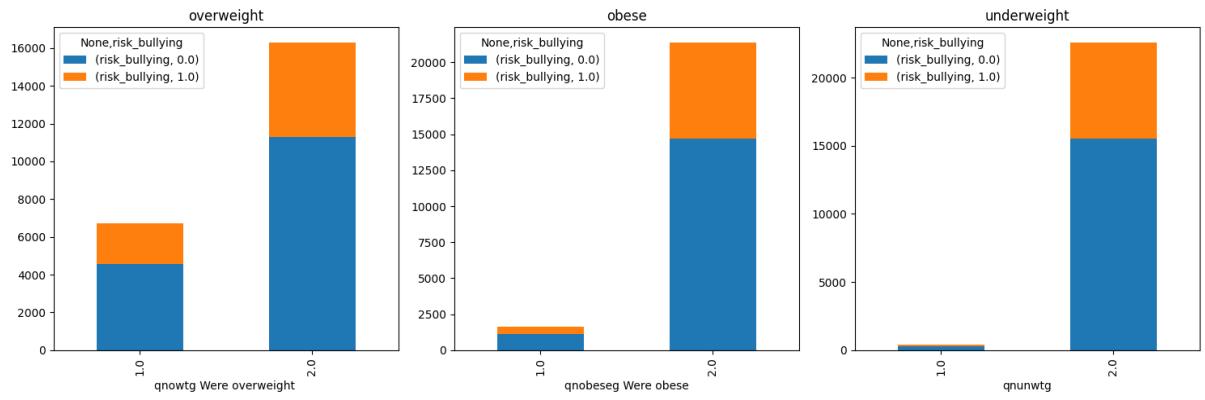


4.2. Gráficas

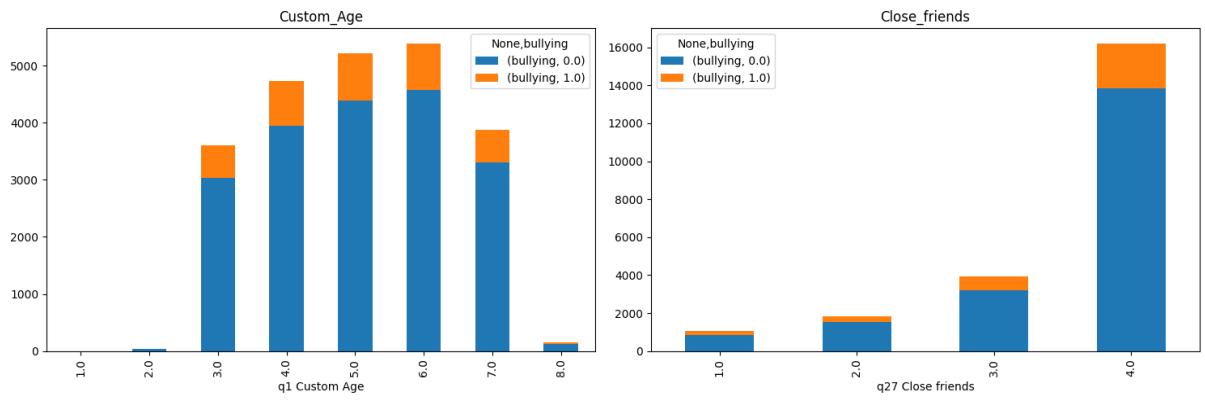
Comparamos en esta gráfica los alumnos que presentan bullying frente a dos variables, edad y número de amigos:



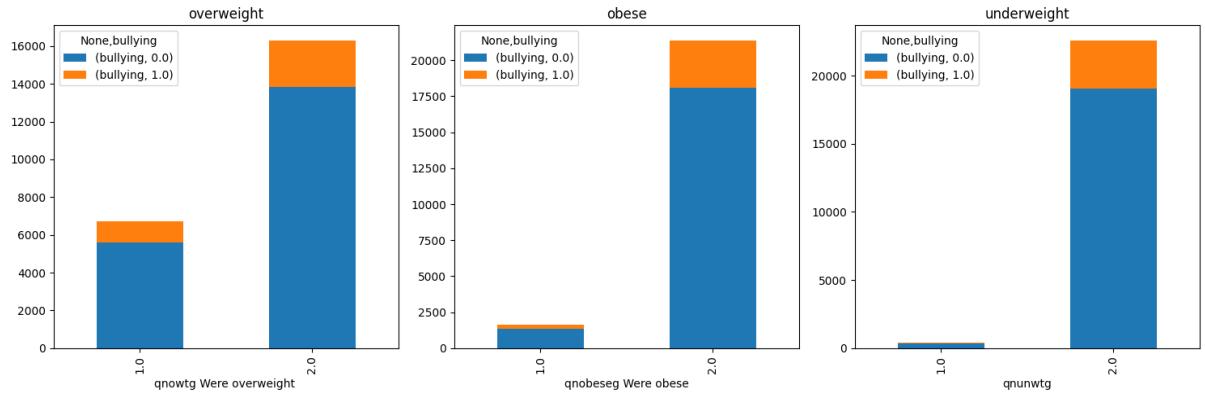
Comparamos en esta gráfica los alumnos que presentan bullying frente a algún problema de peso



Gráfica que representa los casos con ataque físico y dos variables, la edad y el número de amigos:



Gráfica que representa los casos con ataque físico y algún tipo de problema de peso:



5. Resolución de la problemática

5.1. Análisis de datos

Después de la eliminación de las columnas con datos repetidos, el conjunto de datos resultante se compone únicamente de valores numéricos, lo que ha simplificado significativamente su posterior análisis y tratamiento.

Durante el desarrollo del proyecto, se ha prestado especial atención a los valores nulos presentes en el conjunto de datos. Para abordar esta problemática, se han llevado a cabo dos enfoques distintos. En primer lugar, se ha identificado y eliminado aquellas columnas que contenían un porcentaje significativo de valores nulos, así como aquellas columnas que no eran relevantes para el análisis. Una vez realizadas estas acciones, se han aplicado dos técnicas diferentes para tratar con los valores nulos restantes.

En una primera instancia, se ha optado por eliminar todas aquellas filas que contenían valores nulos. Con este dataset reducido se han desarrollado diferentes modelos. El tamaño resultante de este dataset es de (19468, 51).

Posteriormente, se ha utilizado la técnica de KNNImputer para mejorar los resultados del modelo. En primer lugar, se han eliminado los valores nulos de las variables objetivo. A continuación, se ha revisado el número de valores nulos por fila y se han eliminado aquellas filas que contenían más de la mitad de valores nulos. Finalmente, se ha aplicado la técnica de KNN para llenar los valores nulos restantes en el dataset. El dataset resultante de este proceso es de la siguiente forma (54957, 51)

En nuestro estudio, hemos calculado la matriz de correlación entre la variable objetivo y las variables explicativas incluidas en el modelo. La matriz de correlación nos muestra cómo están relacionadas las variables entre sí, proporcionando información sobre la fuerza y la dirección de la relación.

Columns	q15 Physically attacked
q1Custom Age	-0.019534
q2 Sex	-0.050956
q3 In what grade are you	-0.041005
q4 How tall are you	0.012233
q5 How much do you weigh	0.014791
q6 How often went hungry	0.150686
q10 Fast food eating	0.026039
q15 Physically attacked	1.0
q16 Physical fighting	0.329178
q17 Seriously injured	0.147648
q18 Serious injury type	0.084781
q19 Serious injury cause	0.086799
q22 Felt lonely	0.168430
q23 Could not sleep	0.152071
q24 Considered suicide	-0.180886
q25 Made a suicide plan	-0.167595
q26 Attempted suicide	0.203462
q27 Close friends	-0.054230
q28 Initiation of cigarette	0.089794
q29 Current cigarette use	0.130601
q34 Initiation of alcohol u	-0.001059
q35 Current alcohol use	0.109991
q36 Drank 2+ drinks	0.096168
q37 Source of alcohol	0.044000
q38 Really drunk	0.109774
q39 Trouble from drinking	0.143643
q40 Initiation of drug use	0.103620
q41 Ever marijuana use	0.108515
q42 Current marijuana use	0.099924
q43 Amphetamine or methamp	0.077148
q44 Ever sexual intercourse	-0.084652
q45 Age first had sex	0.045855
q46 Number of sex partners	0.099028
q47 Condom use	0.086755
q48 Birth control used	0.079678
q49 Physical activity past	0.024616
q50 Walk or bike to school	0.049611
q51PE attendance	0.039787
q52 Sitting activities	0.046926
q53 Miss school no permissi	0.087591
q54 Other students kind and	-0.086833
q55 Parents check homework	-0.046725
q56 Parents understand prob	-0.104113
q57 Parents know about free	-0.124643
q58 Parents go through thei	0.107934
qnunwtg	-0.006289
qnowtg Were overweight	-0.022703
qnobeseg Were obese	-0.021961

5.2. Preprocesado

En la sección previa, se ha señalado que todos los datos en nuestro conjunto de datos son numéricos, lo que ha simplificado el proceso de preprocesamiento de datos. En todos los modelos, hemos creado una nueva columna que sirve como la variable objetivo. La variable objetivo depende de diferentes columnas en función del modelo. Para el modelo final, la variable objetivo depende de los datos en la columna 'q15 Physically attacked'. En los otros modelos, la variable objetivo depende de las columnas 'qn66 Bullied on school property in past 12 months', 'qn67 Bullied not on school property in past 12 months', y 'qn68 Cyber bullied in past 12 months'.

Para construir la variable objetivo del modelo final, utilizamos el siguiente código:

```
for i in range(len(df)):  
    if df.loc[i, 'q15 Physically attacked'] == 1:  
        df.loc[i, 'bullying'] = 0  
    if df.loc[i, 'q15 Physically attacked'] > 1:  
        df.loc[i, 'bullying'] = 1
```

Después de construir la variable objetivo, escalamos los datos de entrada x utilizando el objeto scaler creado con `MinMaxScaler(feature_range=(0, 1))`. Este escalado comprime los valores de x en un rango entre 0 y 1, lo que puede mejorar el rendimiento del modelo final

5.3. Modelado

5.3.1. Explicación del Modelo

En nuestro proyecto, nos enfrentamos a un problema de desbalanceo en los datos del conjunto de entrenamiento, lo que significa que una o varias de las clases a clasificar tienen muy pocos ejemplos. Para abordar este problema, se aplicaron diversas técnicas con el objetivo de equilibrar las clases y mejorar la capacidad de los modelos de clasificación para detectar correctamente las clases minoritarias.

Para equilibrar el conjunto de datos, se aplicaron tres técnicas diferentes. En la primera, se han eliminado filas aleatoriamente del conjunto de datos para reducir el número de ejemplos de la clase mayoritaria, mediante funciones del tipo:

```
def Balance2(df):  
    cont=0  
    cont2=0  
    cont3=0  
  
    for index, row in df.iterrows():  
        if cont < 30000 and row['bullying']==0 :  
            df = df.drop(index)  
            cont+=1  
        if cont2 < 5000 and row['bullying']==2 :  
            df = df.drop(index)  
            cont2+=1  
        if cont3 < 5000 and row['bullying']==1 :  
            df = df.drop(index)  
            cont3+=1
```

```

        df = df.drop(index)
        cont2+=1
    if cont3 < 3000 and row['bullying']==3 :
        df = df.drop(index)
        cont3+=1
    return df

```

Otra de las técnicas aplicadas ha sido generar un modelo ensamblado mediante dos modelos, en el cual en uno de los modelos se le daba un peso mayor a la variable objetivo menos frecuente, en nuestro caso 1, para que el modelo fuera aprendiendo mejor a la detección de esta variable mediante la función:

```

# Definir la función de pérdida ponderada
from keras import backend as K
# Definir pesos de clases
class_weight = {0: 1, 1: 5}
def weighted_binary_crossentropy(y_true, y_pred):
    # Pesos de las clases
    class_weights = K.constant([1, 5]) # Dar un peso mayor a
    la clase 1

    # Función de pérdida
    y_pred = K.clip(y_pred, K.epsilon(), 1-K.epsilon())
    weighted_cross_entropy = -(y_true * K.log(y_pred) *
    class_weights)
    return K.mean(weighted_cross_entropy, axis=-1)

```

En la tercera técnica, se aplicó la técnica de muestreo "sampling_strategy" para generar ejemplos sintéticos de la clase minoritaria.

La técnica de generación de casos a través de la estrategia de muestreo se utilizó en el modelo final, lo que permitió equilibrar el conjunto de datos y mejorar el rendimiento del modelo en la detección de las clases minoritarias.

En la realización del proyecto se han llevado a cabo varios modelos hasta llegar al modelo final. En primer lugar, se plantearon diferentes variables objetivo utilizando el dataset disponible, tales como 'qn66 Bullied on school property in past 12 months', 'qn67 Bullied not on school property in past 12 months' y 'qn68 Cyber bullied in past 12 months'. Se crearon diferentes modelos para identificar cada una de las clases, mediante la creación de una nueva columna para la variable objetivo, cuyo resultado dependía de los casos positivos de las anteriores variables. Se establecieron diferentes formas de definir la nueva columna objetivo en función de los modelos creados, por ejemplo, en uno de los modelos se establecieron los valores 0 para 'no hay bullying', 1 para 'bullying' en

el colegio', 2 para 'bullying fuera del colegio' y 4 para 'cyber bullying'. No obstante, los resultados obtenidos con estas variables objetivo no fueron satisfactorios, ya que los modelos presentaron bajas sensibilidades.

Posteriormente, se agruparon las variables de "qn66 Bullied on school property in past 12 months" y "qn67 Bullied not on school property in past 12 months" para generar un mismo valor. Otro de los modelos creados se construyó la variable objetivo únicamente con dos valores: 0 y 1. El valor 1 indicaba que no había ningún tipo de bullying y el valor 1 indicaba que se presentaba alguno de los valores anteriores positivos.

Asimismo, se realizaron modelos según el género, con el objetivo de detectar posibles diferencias entre hombres y mujeres, pero los resultados obtenidos no tuvieron una buena sensibilidad.

Finalmente, se optó por realizar un modelo para identificar los casos en los que el bullying se hacía patente mediante un ataque físico al alumno, utilizando la variable objetivo 'q15 Physically attacked'. En este caso, si se producía un ataque físico se consideraba que se estaba presentando bullying, mientras que si no se producía no se consideraba que hubiera bullying. Este modelo presentó la mayor sensibilidad de los calculados y, por lo tanto, se eligió como modelo final para el proyecto. En los modelos anteriores de detección de bullying se ha observado que los resultados obtenidos no son satisfactorios debido a las limitaciones del cuestionario utilizado. En este sentido, se ha identificado que las preguntas formuladas en el cuestionario pueden no ser las más adecuadas o no proporcionar información suficiente para una clasificación precisa de los diferentes grados de bullying. Asimismo, es posible que sea necesario incluir más preguntas o variables en el cuestionario para obtener una evaluación más completa y precisa de la situación de bullying.

Vamos a detenernos a explicar el modelo final.

La variable objetivo es: q15 Physically attacked

Esta columna tiene diferentes valores:

During the past 12 months, how many times were you physically attacked?

- | | |
|---|------------------|
| 1 | 0 times |
| 2 | 1 time |
| 3 | 2 or 3 times |
| 4 | 4 or 5 times |
| 5 | 6 or 7 times |
| 6 | 8 or 9 times |
| 7 | 10 or 11 times |
| 8 | 12 or more times |
| | Missing |

Unifican los resultados de la pregunta del formulario en una sola columna, que toma el valor de 1 si se reporta al menos un ataque físico y el valor de 0 en caso contrario. De esta manera, se obtiene un problema de clasificación binaria,

donde el objetivo es detectar si un estudiante está siendo víctima de bullying físico o no. Aunque esta variable objetivo es más sencilla que las anteriores, se ha demostrado que es más efectiva para detectar los casos de bullying, por lo que se considera adecuada para el propósito del proyecto.

El modelo final es una red neuronal que utiliza la biblioteca Keras de Python. La red consta de varias capas densas que se encargan de procesar la información de entrada, cada una de las cuales tiene un número diferente de neuronas y una función de activación diferente. En este caso, las funciones de activación utilizadas son la función relu y la función sigmoidal.

La capa de aplanamiento (Flatten) convierte los datos de entrada en un vector unidimensional, lo que permite que se puedan procesar de manera más eficiente en la capa final. La capa final es una capa densa con dos neuronas, ya que estamos tratando con un problema de clasificación binaria (1 o 0).

La capa de Dropout se utiliza para reducir el sobreajuste en el modelo. El sobreajuste se produce cuando el modelo se ajusta demasiado a los datos de entrenamiento y no generaliza bien a datos nuevos. La capa de Dropout desactiva aleatoriamente un porcentaje de las neuronas en cada paso de entrenamiento, lo que ayuda a prevenir el sobreajuste.

El modelo se compila con la función de pérdida "categorical_crossentropy", que es una medida de la diferencia entre las predicciones del modelo y las etiquetas verdaderas. El optimizador Adam se utiliza para ajustar los pesos del modelo durante el entrenamiento.

El modelo se entrena con los datos de entrenamiento y validación proporcionados. Durante el entrenamiento, el modelo ajusta sus pesos para minimizar la función de pérdida. Después del entrenamiento, se evalúa el modelo utilizando los datos de prueba para medir su capacidad para generalizar a datos nuevos.

El esquema del modelo final es el siguiente:

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 512)	24576
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 128)	32896
dense_3 (Dense)	(None, 64)	8256
dropout (Dropout)	(None, 64)	0
dense_4 (Dense)	(None, 32)	2080
flatten (Flatten)	(None, 32)	0

```
dense_5 (Dense)           (None, 2)          66
```

```
=====
Total params: 199,202
Trainable params: 199,202
```

Se ha llevado a cabo la selección de hiperparámetros mediante una búsqueda exhaustiva a través del método GridSearchCV. Esta técnica consiste en una búsqueda sistemática de combinaciones de valores para los parámetros de un modelo de aprendizaje automático, con el objetivo de encontrar la configuración óptima que maximice su rendimiento en un conjunto de datos dado.

En concreto, se ha utilizado GridSearchCV para buscar la mejor combinación de hiperparámetros para el modelo de red neuronal desarrollado. Los parámetros en los que se ha realizado la búsqueda son aquellos que afectan directamente al rendimiento del modelo, como la tasa de aprendizaje, el número de neuronas en cada capa, la función de activación, la regularización y el tamaño del lote de entrenamiento.

El proceso de búsqueda exhaustiva se ha llevado a cabo de manera sistemática, evaluando todas las combinaciones posibles de los parámetros seleccionados dentro de un rango de valores predeterminado. De esta manera, se ha logrado encontrar la configuración óptima que maximiza la precisión del modelo en los datos de validación.

5.3.2. Evaluación del modelo

La precisión obtenida con el modelo es: 80.48%

La matriz de confusión del modelo es:

Confusion matrix:

```
[[5473 1355]
 [1323 5565]]
```

Classification report:

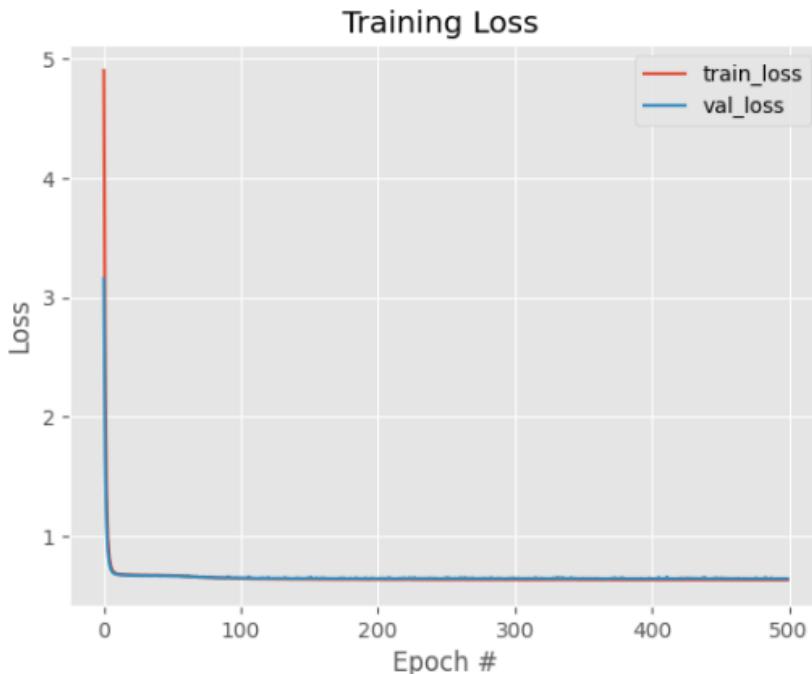
	precision	recall	f1-score	support
0	0.81	0.80	0.80	6828
1	0.80	0.81	0.81	6888
accuracy			0.80	13716
macro avg	0.80	0.80	0.80	13716
weighted avg	0.80	0.80	0.80	13716

Nuestra matriz de confusión muestra que el modelo predijo correctamente 5473 casos negativos (0) y 5565 casos positivos (1), pero también cometió 1355 falsos positivos y 1323 falsos negativos.

El informe de clasificación proporciona una medida de la precisión del modelo, el recall y el f1-score. La precisión se refiere a la proporción de verdaderos positivos entre los casos positivos predichos. El recall se refiere a la proporción de verdaderos positivos entre todos los casos positivos reales. El f1-score es una medida de la precisión y el recall que proporciona una puntuación única que resume el rendimiento del modelo.

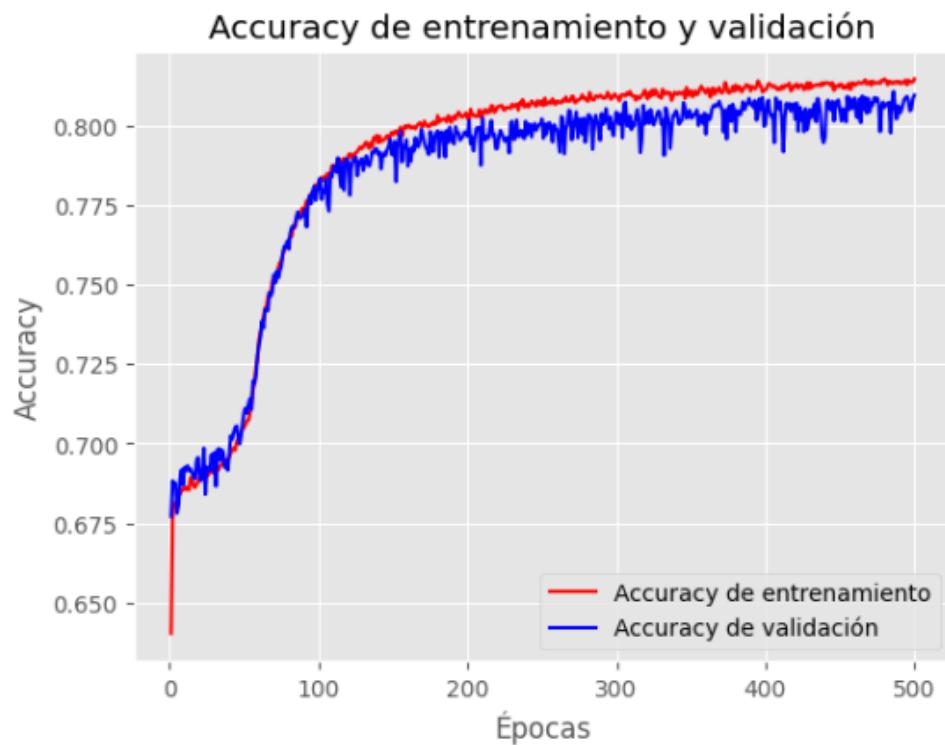
En este caso, el modelo tiene una precisión del 80% para ambas clases y un recall del 80% para la clase 0 y del 81% para la clase 1. Esto significa que el modelo es capaz de predecir correctamente el 80% de los casos para ambas clases. El informe de clasificación también muestra que el modelo tiene un f1-score del 0.80 para ambas clases, lo que indica que el modelo tiene un buen equilibrio entre la precisión y el recall para ambas clases.

La gráfica de las pérdidas del modelo de la red neuronal durante el entrenamiento y la validación:



La gráfica representa dos líneas, una que muestra la pérdida durante el entrenamiento (train_loss) y otra que muestra la pérdida durante la validación (val_loss). La línea de entrenamiento muestra la cantidad de pérdida que se produjo durante el entrenamiento, mientras que la línea de validación muestra la cantidad de pérdida que se produjo en el conjunto de datos de validación. Como se puede observar en la gráfica anterior, no se evidencia overfitting en este modelo, lo que permite concluir que se ha implementado una técnica de análisis adecuada.

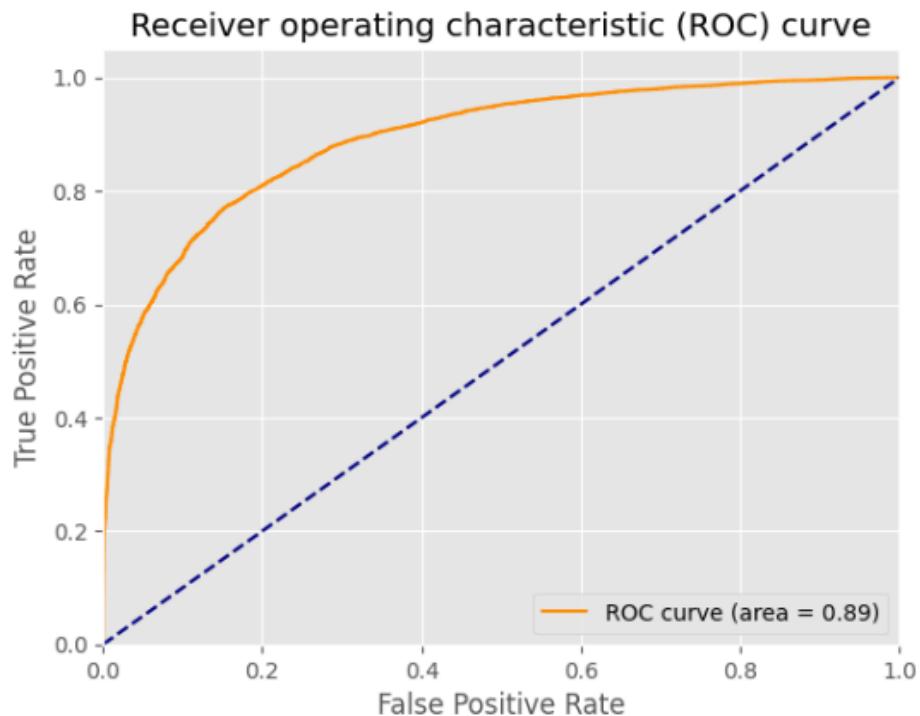
Evolución del accuracy:



La gráfica representa la evolución del accuracy durante el entrenamiento y la validación del modelo. En el eje horizontal se encuentra la cantidad de épocas (iteraciones del modelo) y en el eje vertical se muestra el valor del accuracy obtenido en cada época tanto para el conjunto de entrenamiento (en rojo) como para el conjunto de validación (en azul).

La gráfica permite visualizar la evolución del modelo durante el entrenamiento, verificando si hay algún problema de overfitting en el modelo. Si el accuracy en el conjunto de entrenamiento es alto y el de validación es bajo, se puede estar en presencia de un sobreajuste, lo cual indicaría que el modelo está memorizando los datos de entrenamiento y no generalizando bien para nuevos datos. Si, por el contrario, el accuracy en ambos conjuntos es bajo, entonces se podría estar en presencia de un overfitting, lo que indicaría que el modelo es demasiado simple y no puede capturar la complejidad de los datos. En nuestro caso, se observa que la gráfica es adecuada y no se presenta overfitting.

La curva ROC de nuestro modelo es la siguiente:



La curva ROC representa la tasa de verdaderos positivos (TPR) en el eje y y la tasa de falsos positivos (FPR) en el eje x. Un modelo con un buen rendimiento tendrá una curva ROC cercana al borde superior izquierdo del gráfico (TPR cercana a 1 y FPR cercana a 0), lo que indica una alta tasa de verdaderos positivos y una baja tasa de falsos positivos. El área bajo la curva (AUC) es un valor numérico que indica el rendimiento general del modelo, siendo un valor de 1 el rendimiento perfecto y 0.5 el rendimiento aleatorio. El valor de 0.89 obtenido para el área bajo la curva (AUC) de la curva ROC indica que el modelo de clasificación tiene una buena capacidad para distinguir entre las clases positivas y negativas. Cuanto mayor es el valor del AUC, mejor es la capacidad de un modelo para distinguir entre clases positivas y negativas.

6. Despliegue en la nube

6.1. Información general

Este proyecto tiene como objetivo crear un pipeline para procesar datos de encuestas realizadas a alumnos y predecir si están siendo víctimas de acoso escolar (bullying) o no.

El pipeline se encarga de leer los datos de entrada desde un archivo en formato CSV, realizar una limpieza y transformación de los datos, y luego entrenar y testear un modelo de aprendizaje automático para generar las predicciones.

Para ello se ha utilizado Apache Beam, un modelo de programación de datos unificado que permite el procesamiento de grandes conjuntos de datos de manera eficiente y escalable. Además, se ha utilizado Python como lenguaje de programación, lo que permite un fácil acceso a diversas bibliotecas de aprendizaje automático y análisis de datos.

La idea de este pipeline, es poder ejecutarlo tanto en local como en GCP, mediante DataFlow y AI Platform.

6.2. Estructura de carpetas

El proyecto tiene la siguiente estructura de ficheros:

```
Model_gcp/
|--- __init__.py
|--- preprocess.py
|--- trainer.py
|--- predict.py
|--- retrainer.py
|--- setup.py
|--- .env
|--- requirements.txt
|--- core/
    |--- config.py
```

- `__init__.py`: indica que la carpeta es un paquete de Python.
- `preprocess.py`: script que contiene funciones para el preprocesamiento de los datos de entrada del modelo.
- `trainer.py`: script que contiene la lógica del entrenamiento del modelo de predicción.
- `setup.py`: fichero en donde se define los metadatos y las dependencias del proyecto de Python. Se incluye info sobre el proyecto
- `predict.py`: script que contiene la lógica para hacer predicciones con el modelo entrenado.

- retrainer.py: script que se encargaría de reagrupar los datasets y reentrenar el modelo
- .env: archivo de configuración que contiene variables de entorno.
- requirements.txt: archivo que contiene una lista de dependencias del proyecto.
- core/config.py: archivo que contiene la configuración principal del proyecto, como los parámetros de entrenamiento, los directorios de entrada/salida y las credenciales de acceso a los servicios de Google Cloud.

6.3. Pasos generales del despliegue

6.3.1. procesamiento de datos

Esta es una pipeline de Apache Beam que realizará todo el preprocesamiento necesario para entrenar un modelo de Deep Learning. Utiliza tf.Transform , que es parte de TensorFlow Extended , para realizar cualquier procesamiento que requiera un pase completo sobre el conjunto de datos.

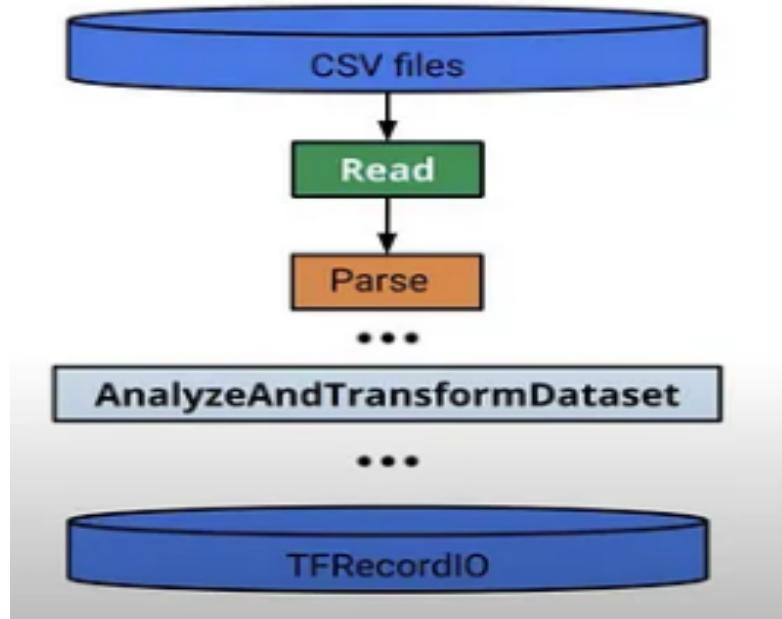
Puesto que entrenaremos una red neuronal para realizar las predicciones, siempre es una buena idea normalizar las entradas a un rango pequeño (normalmente de 0 a 1).

Para realizar este tipo de normalizaciones es necesario revisar todo el conjunto de datos para encontrar los recuentos mínimo y máximo. Afortunadamente, tf.Transform se integra con nuestra canalización de Apache Beam y lo hace por nosotros.

Este fichero tendrá dos flujos de trabajo dependiendo si se trata de un conjunto de train o un conjunto de test.

En el caso de que se trate de un set de train el pipeline consta de varios pasos:

- Lectura de del csv
- Transformación de datos a un diccionario teniendo en cuenta el esquema registrado de las preguntas/respuestas registrado en la fichero core/config.py
- Validación de los datos
- Eliminar filas que puedan contener datos nulos
- Convertimos los datos a float
- Escalado y almacenado del transform(que luego utilizaremos para escalar los datos de test)
- Split de los datos para obtener un conjunto de evaluación
- Almacenar los datos de train y eval



En el caso de que se trate del conjunto de test:

- Lectura de del csv
- Transformación de datos a un diccionario teniendo en cuenta el esquema registrado de las preguntas/resuestas registrado en la fichero core/config.py
- Validación de los datos
- Eliminar filas que puedan contener datos nulos
- Convertimos los datos a float
- Descarga del transform alojado en el bucket y escalado de los datos
- Almacenar conjunto de test

6.3.2. Entrenando al modelo (training.py) (sin implementar)

Entrenaremos una red neuronal profunda a través de TensorFlow. Este apartado del proyecto utilizará los datos almacenados en el directorio de trabajo. Durante la etapa de preprocesamiento, tf.Transform generó un gráfico de operaciones para normalizar los datos.

Para conjuntos de datos pequeños, será más rápido ejecutarlo localmente. Si el conjunto de datos de entrenamiento es demasiado grande, se escalará mejor para entrenar en AI Platform .

6.3.3. Predicciones (predict.py) (sin implementar)

En esta fase del proyecto, podríamos optar por dos tipos de predicciones, por lotes o en stream. Puesto que las necesidades del proyecto no exigen una

predicción inmediata, podríamos optar por una predicción por lotes, mejorando el rendimiento a costa de sacrificar latencia.

6.3.4. **Agrupamiento de datasets y reentrenamiento** (retraining.py) (sin implementar)

Este script, realizará la agrupación del dataset original, con todos los test que se han ido realizando y almacenado en la instancia de postgres, por lo que la idea sería que realizará de forma programada dicha agrupación y posterior entreno, realizando una valoración de los datos del modelo en servicio y del modelo re-entrenado.

6.4. **Tecnologías**

Apache Beam

Apache Beam es un modelo unificado de código abierto para definir canalizaciones por lotes y de procesamiento paralelo de datos de transmisión. El modelo de programación de Apache Beam simplifica la mecánica del procesamiento de datos a gran escala. Con uno de los SDK de Apache Beam, puedes compilar un programa que define la canalización. Luego, uno de los backends admitidos de procesamiento distribuido de Apache Beam, como Dataflow, ejecuta la canalización. Este modelo te permite concentrarte en la composición lógica de los trabajos de procesamiento de datos en lugar de en la organización física del procesamiento paralelo. Puedes enfocarte en lo que necesitas que haga tu trabajo en lugar de en cómo se ejecuta.

Aquí hay algunas razones por las que hemos elegido Apache Beam para realizar un pipeline:

- **Portabilidad:** Apache Beam permite escribir pipelines de procesamiento de datos una vez y ejecutarlos en diferentes plataformas, lo que facilita la migración entre diferentes proveedores de servicios en la nube o la ejecución en diferentes entornos locales.
- **Escalabilidad:** Apache Beam proporciona una capa de abstracción sobre la infraestructura de procesamiento de datos subyacente, lo que permite aprovechar la escalabilidad y el paralelismo ofrecidos por diferentes plataformas de procesamiento de datos.

- **Flexibilidad:** Apache Beam permite definir pipelines de procesamiento de datos flexibles y escalables que pueden manejar diferentes tipos de datos y diferentes fuentes y destinos de datos.
- **Facilidad de uso:** La API unificada de Apache Beam hace que sea fácil de aprender y utilizar, lo que permite a los desarrolladores escribir pipelines de procesamiento de datos de manera más rápida y eficiente.
- **Multi-lenguaje:** Otra ventaja que aporta Apache Beam es la capacidad de que cada runner funciona con cada lenguaje, por lo que se pueden implementar pipelines multi-lenguaje con transformaciones cross-language.

DataFlow

Dataflow es el servicio de procesamiento de datos serverless en Google Cloud Platform (GCP) que permite procesar y analizar grandes cantidades de datos en tiempo real o en batches de manera unificada. Es la solución estándar de ETL en Google Cloud, más moderna y ágil que alternativas como Dataproc.

Dataflow está basado en Apache Beam (proyecto open source que combina procesamiento streaming y batch, de donde viene su nombre) y permite crear flujos de trabajo para procesar, transformar y analizar datos utilizando una variedad de herramientas y lenguajes de programación.

AI-Platform

AI Platform permite entrenar tus modelos de aprendizaje automático a gran escala, alojar tu modelo entrenado en la nube y usar tu modelo con el propósito de realizar predicciones sobre datos nuevos. El servicio de entrenamiento de AI Platform te permite entrenar modelos con una amplia variedad de opciones de personalización diferentes.

Puedes seleccionar numerosos tipos de máquina diferentes para potenciar tus trabajos de entrenamiento, habilitar el entrenamiento distribuido, usar el ajuste de hiperparámetros y acelerar con GPU y TPU.

7. Presentación de resultados

7.1. Suposiciones iniciales. A completar

Con nuestro modelo final, nuestro objetivo era ser capaces de predecir los diferentes grados de bullying y ciberbullying.

Al nivel de la aplicación web queríamos que el centro tuviese credenciales de acceso y que el alumno no necesitase estar logueado para así poder realizar el test desde cualquier lugar, en privado y en confianza. También propusimos que fuese el centro el que generase las claves para tantos alumnos necesitase.

7.2. ¿Cuáles nos han sido válidas? ¿Cuáles no? A completar

Nuestro modelo no ha sido capaz de predecir con precisión los diferentes grados de bullying o cyberbullying debido a que el conjunto de datos que utilizamos no es lo suficientemente representativo o completo para hacer una clasificación precisa. Sin embargo, para predecir los casos graves de bullying, como son aquellos en los que se ha producido un ataque físico, se ha obtenido un modelo preciso. Por lo tanto, se ha simplificado el resultado y se ha centrado en la capacidad del modelo para predecir estos casos graves.

En cuanto a la web se han podido llevar a cabo todas las suposiciones, excepto el despliegue cloud que, aunque está alojada en un bucket y están hechos los balanceadores, no se ha podido hacer que funcione correctamente.

7.3. Métricas seleccionadas y por qué A completar

El modelo que hemos desarrollado es un modelo de clasificación binaria y, por tanto, se han aplicado las siguientes métricas y medidas de evaluación para su evaluación y validación. Dichas medidas se encuentran detalladas en el apartado correspondiente del presente documento dedicado al modelo. Y son las siguientes:

Accuracy: mide la proporción de predicciones correctas en relación con el número total de predicciones realizadas. Junto con la grafica podemos observar si tenemos o no tenemos overfitting

Precisión: mide la proporción de verdaderos positivos (TP) en relación con el número total de predicciones positivas (TP + FP).

Sensibilidad o Tasa de Verdaderos Positivos (Recall o True Positive Rate): mide la proporción de verdaderos positivos (TP) en relación con el número total de casos positivos reales (TP + FN).

Especificidad o Tasa de Verdaderos Negativos (True Negative Rate): mide la proporción de verdaderos negativos (TN) en relación con el número total de casos negativos reales ($TN + FP$).

Valor F1 (F1 Score): combina la precisión y la sensibilidad para proporcionar una medida general del rendimiento del modelo.

Área bajo la curva ROC (AUC-ROC): mide la capacidad del modelo para distinguir entre clases positivas y negativas.

Pérdida logarítmica (Log Loss): mide la discrepancia entre las predicciones del modelo y los valores reales, donde un valor más bajo indica una mejor precisión. Y junto con la gráfica podemos ver si nuestro modelo presenta overfitting.

Las métricas mencionadas anteriormente ofrecen información sobre la capacidad del modelo para predecir los casos de bullying y creemos que está realizando esta tarea de manera efectiva.

7.4. Arquitectura elegida. ¿Ha sido la definitiva?

Desde un principio la arquitectura la hemos desarrollado en Google Cloud Platform. Dicha arquitectura constaba de una BBDD SQL de postgre, una API desarrollada con fastapi y desplegada en App Engine, una web creada con JavaScript, html y CSS, también desplegada en un App Engine y la puesta en producción del modelo con Apache Beam mediante Dataflow y AI-Platform.

Prácticamente ha sido la arquitectura definitiva, aunque la web debía estar funcionando en GCP junto con la API y el modelo, hemos tenido múltiples problemas con los despliegues, principalmente con problemas en versiones de algunas dependencias.

En las siguientes líneas entraremos más a fondo en algunas de las partes del proyecto:

- **API-FastApi**

En cuanto a la elección del FrameWork, desde un principio no tuvimos dudas. Ninguno de los componentes del equipo tenía grandes conocimientos en desarrollo de APIS por lo que la opción de FastApi, sin duda, era la más razonable en cuanto a la curva de aprendizaje se refiere. Si a eso le

añadimos las múltiples ventajas que nos ofrece esta tecnología, no fue una decisión difícil de tomar.

Junto con las librerías peewe y pydantic, hemos desarrollado una API que es capaz de satisfacer todas las peticiones requeridas por la web de una forma óptima y con buen rendimiento.

Uno de los mayores quebraderos de cabeza en esta parte del proyecto, fue diseñar y organizar las necesidades de la web. Desde un principio teníamos claro de la importancia de la confidencialidad, tanto de la información recabada en los test de los alumnos, como de los resultados obtenidos mediante las predicciones. En base a esta premisa tuvimos que realizar múltiples cambios a lo largo del desarrollo para ofrecer un servicio en el que la privacidad del alumno fuera prioritaria y que los datos sensibles fueran responsabilidad del colegio. Para ello creamos un registro de colegios y una autentificación, para que desde su perfil, pudieran solicitar los id en base al nivel, curso y aula del alumno. Esos id son strings en el que añadimos información relativa a la fecha, colegio, nivel, curso, aula y número del alumno. De esta forma pudimos crear un endpoint que no requiere autentificación del colegio, en el que el alumno a través de su id puede acceder al test y realizarlo.

En algunas de las modificaciones y mejoras que fuimos implementando, tuvimos que realizar múltiples pruebas API-WEB para optimizar bien las interacciones entre ambas apps.

Otro problema que hemos tenido, y que ha modificado el endpoint de la predicción dentro de la API, ha sido, como hemos comentado en la parte de despliegue del modelo, el no haber podido realizar la compilación en Dataflow del pipeline. En la idea inicial, teníamos previsto realizar la puesta en producción del modelo en GCP, y almacenar tanto el transform que se encargaría de realizar el escalado de los datos, como el propio modelo en GC Storage. De esta forma la API realizaría la descarga de ambos ficheros para realizar el escalado y la predicción. Este problema nos hizo replantearnos la obtención de los ficheros necesarios para la predicción dentro de la API, de forma que optamos por subir de forma manual tanto el escaler como el modelo al bucket de GCP.

En cuanto al despliegue en GCP, valoramos dos opciones, a través de Kubernetes creando un Dockerfile con la posibilidad de realizar lo mismo con la web, y mediante App Engine. Tras varias pruebas con Kubernetes, decidimos la segunda opción debido a su sencillez.

En cuanto a dicho despliegue de la API en App Engine, hay que comentar que, en las primeras fases del proyecto, conseguimos compilar y hacer funcionar la API sin problema en GCP. Pero a medida que fue creciendo la app y tuvimos que añadir librerías como TensorFlow a las dependencias, nos empezó a dar problemas de versiones, generando conflictos con dependencias que en las primeras compilaciones funcionaban

perfectamente. Debido a la falta de tiempo no lo hemos vuelto a probar, pero creemos que podría funcionar el sustituir tensorflow por tensorflow-gpu.

A pesar de los problemas expuestos, que no tienen mucho que ver con el framework en sí, consideramos que la opción de fastapi ha sido totalmente acertada, no solo por las ventajas comentadas anteriormente, si no porque permite una fácil puesta en producción y además en base al diseño que hemos planteado en la arquitectura de ficheros nos permitirá una fácil escalabilidad en un futuro.

- Despliegue del modelo en GCP

Desde el primer momento que empezamos a desarrollar el despliegue del modelo nos encontramos con problemas. Como ninguno de los integrantes del equipo tenía conocimientos sobre Apache Beam, fue un reto desde el minuto uno. La primera fase de desarrollo fue la creación del pipeline de preprocesado. A pesar de que nuestros datos en bruto no requieren un gran preprocesado, nos surgieron errores desde el principio.

El primer gran problema fue la fase del escalado (MinMaxScaler), ya que esta fase requiere un pase completo sobre el conjunto de datos. En los primeros intentos creamos una clase que realizaba el escalado con la clase de scikit-learn, pero nos devolvía todos los valores a cero ya que no disponía de los datos de máximo y mínimo del dataset para poder realizar el escalado correctamente. Tras investigar en la documentación de Apache Beam y GCP, descubrimos que este procesado se realiza a través de TensorFlow-Transform.

A pesar de que hubo que realizar diferentes parseados de los datos para que funcionara el escalado, no fue muy complicado implementarlo y poner a funcionar en local todo el pipeline.

El siguiente gran problema surgió cuando hicimos el despliegue en DataFlow, tras múltiples problemas con permisos de la cuenta de servicio de GCP, conseguimos arrancar el pipeline en la nube, pero tras el aprovisionamiento de las máquinas el proceso expiraba antes de realizar el preprocesado. El problema fue debido a un problema de versiones con Tensorflow y TensorFlow-transform, que por falta de tiempo aún no pudimos subsanar.

Debido a todos estos problemas que nos fueron robando bastante tiempo, únicamente nos ha dado tiempo a poner en marcha el preprocesado en local, por lo que no hemos podido probar el despliegue del training en AI-Platform.

A pesar de no haber podido avanzar todo lo deseado en el desarrollo de esta parte del proyecto, consideramos que Apache Beam es una buena opción en base a nuestro objetivo, ya que ofrece una buena escalabilidad al proyecto y

además ofrece una fácil migración entre diferentes proveedores de servicios en la nube lo cual con vistas al futuro podría ser interesante.

7.5. **Métodos elegidos. ¿Cuáles han sido los mejores? A completar**

Para obtener el modelo final se utilizó la técnica de deep learning mediante una red neuronal. Esta elección se basó en la cantidad de datos disponibles, ya que se consideró como la forma óptima para desarrollar el modelo.

Además, se realizaron diferentes modelos de machine learning con el fin de poder realizar una comparación entre los datos obtenidos.

En cuanto a la web, la elegimos desarrollar en html y css con javascript por su versatilidad y lo cual ha sido muy buena idea en cuanto a conseguir toda la versatilidad que queríamos que tuviera la aplicación web. Pero teniendo en cuenta precisamente esa dificultad , y que no somos expertos en la materia, nos ha llevado a numerosas horas de programación y quebraderos de cabeza. Aún así, estamos contentos con el resultado y creemos que ha merecido la pena.

7.6. **Retrospectiva ¿Qué haríamos igual?¿Qué cambiaríamos?**

Se recomienda modificar el cuestionario utilizado para recopilar información de los estudiantes, de manera que se puedan obtener más datos relevantes para la predicción del bullying. Es importante incluir preguntas sobre temas de redes sociales, acoso verbal en plataformas como WhatsApp y otras formas de ciberacoso. Esta información adicional permitiría una mejor comprensión del fenómeno del bullying y una posible mejora en el modelo utilizado. Después de obtener los nuevos datos, se deberá realizar una revisión exhaustiva del modelo y, si es necesario, ajustarlo para mejorar su precisión y sensibilidad en la predicción del bullying.

7.7. **Información obtenida del dataset**

Como se ha mencionado anteriormente el principal problema es el cuestionario llenado por los alumnos, es necesario diseñar un nuevo cuestionario que proporcione una mayor cantidad y calidad de datos. Este nuevo cuestionario debería incluir preguntas específicas sobre temas de redes sociales, conversaciones de WhatsApp, grabaciones de videos no consentidas, entre otras.

También en el cuestionario se realizan preguntas como por ejemplo el consumo de tabaco de los padres que creemos que no es relevante en nuestro estudio.

7.8. Conclusiones

Considerando que hemos partido de un dataset desbalanceado y con muchos datos que no eran válidos para predecir se estima haber obtenido un modelo óptimo para la predicción de casos graves de bullying.

Se ha logrado una arquitectura WEB-API-BD dinámica y versátil. Una API escalable, concebida para poder añadir funcionalidades, variedad de contenido y optimizaciones reduciendo enormemente los tiempos de desarrollo y capacitada para mantener el funcionamiento global con modelos predictivos diferentes, incluso está diseñada para poder utilizar diferentes cuestionarios con diferentes modelos añadiendo unas pocas líneas de código y una web desarrollada de forma que se adapta a todos los posibles cambios del proyecto.

8. Demo

En el siguiente enlace podemos ver una demostración de lo que es el proyecto

[Enlace Youtube](#)