

Sistema Acción Tutorial

Daniel Alberto Vazquez Ramirez, Jose de Jesus Rodriguez Hernandez, Edgar Maximiliano Morales Morales, Victor Manuel Zamora Ramos

*CENTRO UNIVERSITARIO DE CIENCIAS
EXACTAS E INGENIERÍAS, (CUCEI, UDG)*

daniel.vazquez7472@alumnos.udg.mx
jose.rodriguez2962@alumnos.udg.mx
edgar.morales5687@alumnos.udg.mx
victor.zamora@academicos.udg.mx

Abstract— Currently, the issue of academic lag and dropout rates at the university center poses a significant challenge stemming from various factors jeopardizing the completion of higher education studies. The development of a tool facilitating the connection between tutors and students, providing monitoring and control in tutoring administration, establishes a robust foundation for mitigating the serious dropout problem. Although there is a variety of software tools available in the market that could support a tutoring program, we have identified that creating a proprietary platform integrating student data visualization and grouping can yield a competitive advantage. Furthermore, if students can also seek clarification through can yield a competitive advantage. Furthermore, if students can also seek clarification through the utilization of a chatbot, it would enhance the platform's efficacy.

Palabras claves – Proyecto modular, tutorías, k means, cloud, redes neuronales, MySQL, AWS.

I. INTRODUCCIÓN

El rezago y deserción escolar es un problema que se vive en todos los niveles escolares, teniendo base en distintos motivos que originan este problema. Actualmente existen diversos apoyos y soluciones que buscan reducir el incremento de la tasa de estudiantes que desertan atendiendo causas específicas, sin embargo, es evidente que cada alumno se encuentra en una situación distinta debido a factores particulares. A lo largo del trayecto universitario hemos tenido la experiencia de compartir el proceso educativo profesional con distintos compañeros y hemos observado cierta problemática recurrente como la falta de información y orientación, así como el apoyo por parte de algún tutor durante la carrera. Analizando lo anterior existe un punto importante del cual se encontró una oportunidad para el desarrollo de un proyecto que permita atender un factor específico de la deserción estudiantil en beneficio de la comunidad universitaria. Sistema Acción Tutorial ha sido pensado como una herramienta que permita la capacidad de establecer relaciones de tutores con alumnos, buscando brindar un acompañamiento a los estudiantes en apoyo a la formación integral, con el objetivo de desarrollar estrategias para evitar el rezago y la deserción escolar del Centro Universitario de Ciencias Exactas e Ingenierías.

A. Planteamiento del Problema

Anteriormente se definió el problema de rezago y deserción escolar y si bien, existen plataformas como Google Classroom, Google Forms y diversas plataformas educativas que otorgan distintas herramientas, estas presentan una generalización en el ámbito y no se centralizan en las necesidades particulares para el correcto seguimiento de tutorías en el contexto universitario.

B. Justificación

El desarrollo de este proyecto enfatiza en los requerimientos específicos del centro universitario atendiendo las necesidades particulares del problema base como la asignación de tutorías y el correcto seguimiento de los procesos agregando como herramienta un chatbot que permita resolver dudas relacionadas a los distintos puntos encontrados en la normativa del Reglamento General para la Prestación del Servicio Social de la Universidad de Guadalajara, Reglamento General de Titulación de la Universidad de Guadalajara y el Reglamento General de Evaluación y Promoción de Alumnos de la Universidad de Guadalajara. En búsqueda de generar valor a la información obtenida del uso de este sistema, se implementa una visualización de información a través de un dashboard y gráficos que muestran la evaluación de la información utilizando un algoritmo de clusterización que permita obtener una visión más amplia de los estudiantes tutorados.

II. TRABAJOS RELACIONADOS

En el ámbito online existen diversas herramientas desarrolladas con fines académicos, las cuales permiten la creación de clases, llevar un control de evaluaciones así como sesiones virtuales que sin duda han sido de gran apoyo para el trabajo actual dentro de las escuelas, así mismo se podría incluir utilidades de distintos softwares para llevar a cabo el seguimiento de tutorías, donde incluso la comunicación a través de correo electrónico podría suponer una solución para establecer relaciones entre tutores y estudiantes. Sin embargo, dado el enfoque del proyecto y el contexto particular que buscamos solucionar, encontramos que el software mencionado anteriormente podría ser una herramienta de apoyo, más no el principal objeto de solución debido a que se busca mantener un

seguimiento particular del alumno con distintas actividades que permitan un desarrollo integral en la universidad, otorgando una visión más amplia de información recopilada sobre las respuestas a los tutores así como brindar una herramienta de apoyo a los alumnos en búsqueda de resolución de dudas específicas de normatividad anteriormente mencionada.

III. DESCRIPCIÓN DEL DESARROLLO DEL PROYECTO MODULAR

A. Hipótesis

La implementación de este proyecto aumentará la eficiencia en la asignación y seguimiento de tutorías en el centro universitario, permitiendo que los estudiantes logren conectarse de manera más sencilla y rápida con un programa de tutorías, atendiendo alumnos con alta probabilidad de deserción o rezago. El chatbot podrá responder preguntas sobre las distintas normativas encontradas en el Reglamento de Servicio Social, Titulación y Evaluación y Promoción de la Universidad de Guadalajara. Finalmente almacenar y recuperar la información recopilada de las actividades mostrarán una visión amplia sobre las respuestas de los estudiantes durante el transcurso de las tutorías lo que permitirá enfocar los esfuerzos de los tutores a ciertas problemáticas encontradas.

B. Metodología

Existen distintas metodologías de trabajo y organización que permiten el desarrollo de manera efectiva de los proyectos, sin embargo, después de analizar el problema y dadas las necesidades que supondría el desarrollo del mismo, decidimos elegir SCRUM debido a la flexibilidad, adaptabilidad, entrega iterativa de valor, colaboración y mejora continua que ofrece. La comunicación constante entre los miembros del equipo se convirtió en un punto esencial para el correcto desarrollo del proyecto e implementación de los módulos correspondientes, además de que representó un gran apoyo a la resolución de los distintos problemas que surgieron, así como cambios durante el tiempo. El proyecto estuvo bajo las distintas fases que lograron el éxito permitiendo el avance por la planificación, sprints, sesiones semanales para la revisión de avances y problemas, utilizando herramientas como Google Meet y GitHub.

Por parte de las tecnologías elegidas para este proyecto incluyeron el framework Django dado que utiliza el patrón de diseño conocido como modelo vista controlador y para la interacción con la base de datos el patrón Object Relational Mapping por lo que nos otorga herramientas que facilitan y aceleran el desarrollo estableciendo el lenguaje python para el backend y sobre el frontend trabajando con html, javascript con apoyo para el diseño de CSS. El almacenamiento de datos se implementó mediante MySQL utilizando un contenedor Docker. El desarrollo dadas las características del framework inició con la creación de los modelos, es decir las distintas tablas con las que trabajaremos, para esto nuestro análisis y diagrama de entidad relación ya se encontraba finalizado por lo que resultó sencillo el cumplir la primera fase. Es importante mencionar que estos modelos permiten el manejo de diferentes

tipos de datos. La siguiente fase implicó el trabajo constante debido a la comunicación entre las vistas y los controladores, en este caso Django lo maneja mediante views y templates, por lo que para mostrar información y visualizar vistas se requiere de manera obligatoria la implementación de las vistas. En la planeación del proyecto se optó por dividir el trabajo y esfuerzo en diferentes sprints a lo largo del desarrollo. Para garantizar el funcionamiento del sistema se llevaron a cabo pruebas unitarias, de integración y de usabilidad.

C. Diseño y fase del prototipado del proyecto modular

Derivado de la planificación, análisis del problema, establecimiento de objetivos, definición de alcance del proyecto entre otros puntos esenciales, fue como comenzamos la etapa del diseño. El desarrollo del esquema de la base de datos, así como el diccionario de datos fue una de las claves esenciales para el éxito del proyecto, considerando primordialmente la posibilidad de escalabilidad del mismo. Para el esquema se contemplaron entidades como usuarios, tutorías, reportes, actividades, sesiones y coordinaciones principalmente, sin embargo, al momento de analizar las distintas transacciones dentro de la base de datos generaron tablas que no se tenían contempladas tales como calendario de convocatorias, calendario de periodo, perfiles de usuario. Se aprovecharon herramientas que proporciona el framework de Django como el sólido sistema de gestión de usuarios integrado, que incluye autenticación, autorización y otras características relacionadas, por lo que esto modifica el esquema inicial y constituye la creación de los modelos de perfiles, así como el uso de los grupos de usuarios para el manejo de las vistas y permisos. Esto representa una ventaja para el avance del proyecto ya que las sesiones las administra el mismo framework y el almacenamiento de contraseñas las realiza mediante el algoritmo de PBKDF2 (Password-Based Key Derivation Function 2) con un algoritmo de hash SHA256.

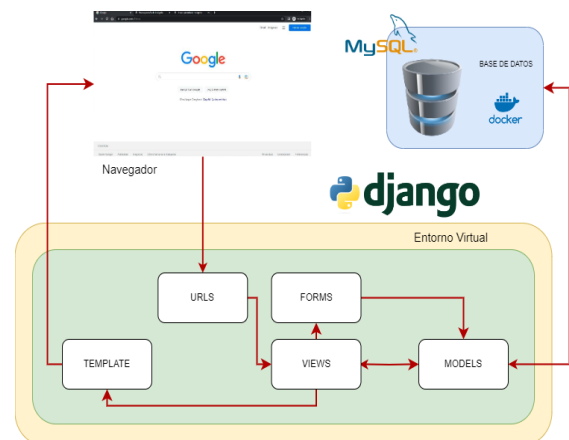


Fig. 1. Esquema que demuestra el funcionamiento del proyecto en desarrollo con el framework Django. Elaboración propia.

El esquema final obtenido mostrado en la Fig. 1 muestra las relaciones existentes entre todas las entidades del proyecto, para ello se implementaron los distintos modelos a través del

framework para posteriormente aplicar las migraciones correspondientes para que la base de datos se construyera de una manera más sencilla y rápida. Durante la etapa de diseño y desarrolló se utilizó una base de datos MySQL dentro de un contenedor Docker y como anteriormente se mencionó, el uso de las herramientas del ORM de Django.

El modelo de desarrollo de software utilizado fue la arquitectura monolítica, aunque el mismo framework da la posibilidad de sensación de trabajo a través de microservicios debido a la capacidad de dividir el proyecto en diferentes apps, lo que permite clasificar las funcionalidades separadas, aunque dependen de funcionar dentro del mismo entorno.

Esto permitió el avance y trabajo independiente ya que alguna modificación dentro de la app de gestión de tutorías no afectaría a los usuarios o creación de reportes.

En el apartado anterior de metodología se presentaba de forma general el funcionamiento del frontend utilizando el patrón modelo, vista y controlador, por lo que el desarrollo representó una dependencia entre los distintos componentes, por lo tanto, decidimos dejar establecida la base de datos con anterioridad basándonos en el siguiente fundamento. Django nos proporciona un desarrollo mediante formularios los cuáles mantienen una conexión directa con los modelos, las vistas utilizan los formularios que finalmente renderiza en las plantillas o documentos HTML y para el acceso por el navegador necesita de la implementación mediante el módulo de urls que conectan con las vistas, lo que en consecuencia provocó el desarrollo de los distintos sprints mediante la construcción y conexión de los anteriores componentes constantemente. El equipo analizando lo anterior decide

manejar el proyecto de la siguiente manera: interacción con la base de datos a través de los modelos, utilizar la generación de formularios del framework, las vistas se encargan de gestionar la información que mostrará al usuario, las plantillas mediante documentos HTML, diseño a través de CSS y JavaScript solo para la interacción con el usuario. Antes de comenzar con la codificación de las vistas de usuario se realizaron distintos mockups que permitieron establecer estándares y el estilo que tendría el sistema. Los sprints iniciales se dedicaron a la gestión de usuarios a través de distintos perfiles y de grupos que permiten el acceso e interacción dentro del software. Para limitar la creación de cuentas y de administrar los grupos, se decidió por aceptar solo dominios como “@academicos.udg.mx” y “@alumnos.udg.mx” lo que mediante expresiones regex nos facilitó la identificación de los usuarios (tutores y estudiantes) así como la validación de correos mediante el envío de código de verificación a dichos correos durante el registro en el sistema.

Continuando con el desarrollo se crearon las bases de los distintos usuarios avanzando a través de las distintas funcionalidades del sistema teniendo entonces para tutores la consulta de tutorías, generación de reportes a estudiantes, creación de sesiones, consulta de respuestas de actividades, solicitar reasignación de tutorías al igual como una posible baja; los estudiantes podrán solicitar tutorías (dentro del periodo de convocatoria), consultar coordinaciones, consultar tutores, contestar formularios de actividades, registrar reportes a tutores, consultar sesiones y el acceso al chatbot; finalmente el usuario administrador del sistema podrá realizar la consulta y asignación de tutorías, así como la cancelación o reasignación,

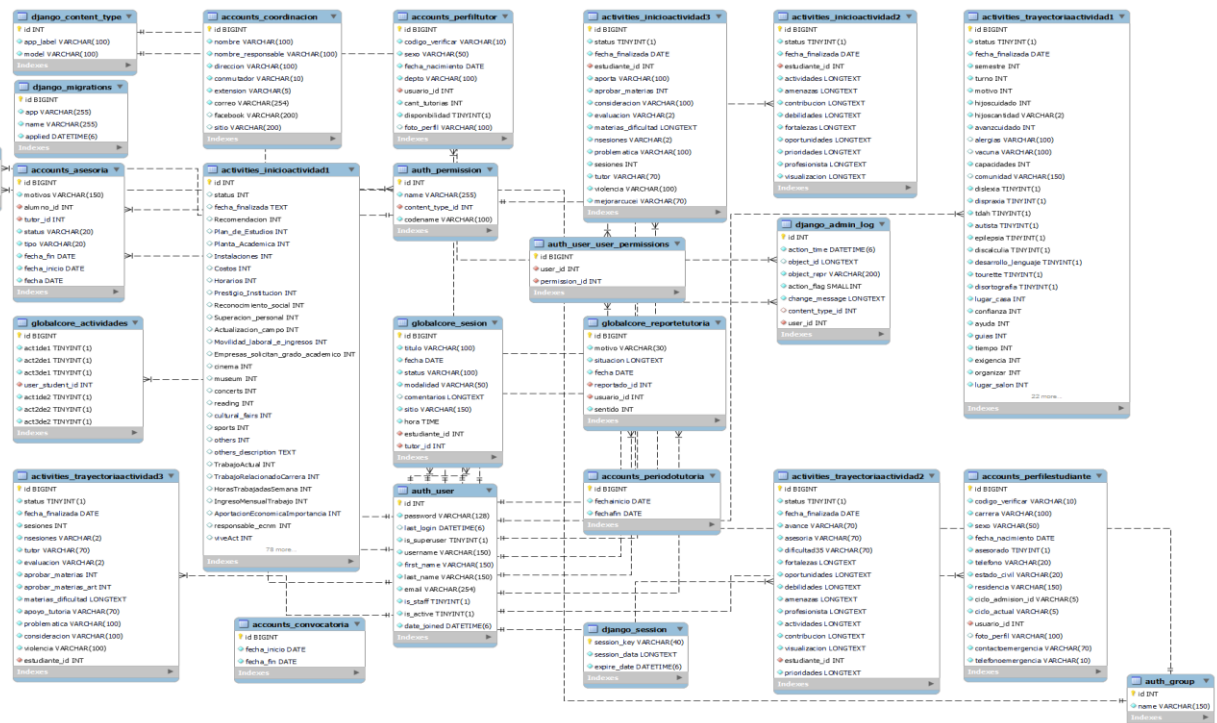


Fig.2. Diagrama ER de la base de datos. Elaboración propia.

consultar información de tutores y estudiantes, así como sus respectivos reportes y actividades para el caso de los alumnos, actualización de calendarios, visualización de clusterización y estadísticas de información recopilada derivada de la contestación de actividades.

Para la gestión de tutorías se involucran los estudiantes mediante la solicitud de la misma dentro del periodo establecido por el administrador, este último es quien asigna los tutores a dicho estudiante o en su caso tiene facultad de cancelar la solicitud derivado de causas como tutores no disponibles o alguna solicitud realizada por el alumno. El tutor simplemente podrá observar dentro del panel las tutorías asignadas. Buscando facilitar la visualización de tutorías para el administrador, por default se muestran las que tienen status de “solicitada” sin embargo, existe la opción de filtrar por los diferentes status almacenados. El estudiante recibirá mediante correo electrónico el aviso de solicitud de tutoría, así como la notificación de asignación de tutoría con los datos sobre la misma o en su caso los datos de cancelación. Un punto importante a mencionar es el funcionamiento de envío de correos y para esto Django proporciona una biblioteca llamada “django.core.mail” y utilizamos el servidor SMTP de Gmail junto con una cuenta dedicada a la administración del sistema, además de la habilitación de acceso de aplicaciones y generación de contraseñas para la correcta conexión. El siguiente módulo desarrollado fue el registro de reportes, para ello los estudiantes podrán generar reportes a sus tutores al igual que de forma inversa, justificado por motivos válidos y un

campo de texto donde se podrá expresar alguna situación dada para que el administrador tenga la capacidad de seguimiento ante problemas durante la gestión de alguna tutoría. El administrador es notificado mediante correo electrónico para su futura consulta del reporte dentro de plataforma. Es esencial el control de esta funcionalidad, por lo que decidimos manejar un status de la tutoría que permita el acceso o restricción, permitiendo solo el registro mientras la tutoría se encuentre en curso. El siguiente módulo se dedica a la gestión de sesiones o reuniones con el fin de llevar un registro sobre los acercamientos de tutores y estudiantes a lo largo del periodo de las tutorías, para ello el tutor solo podrá crear sesiones y actualizar, es decir la posibilidad de finalizar con éxito o cancelarla por razones ajenas.

Un punto esencial que le brinda las herramientas a este proyecto son las actividades que el estudiante podrá realizar durante el periodo de su tutoría y que mediante una sesión el tutor puede dialogar y trabajar con su tutorado. Este proyecto se basa en 3 diferentes tutorías: inicio, trayectoria y egreso. La tutoría de inicio refiere a los primeros tres semestres donde el alumno vive el periodo de integración a la vida universitaria, aquí se busca entrevistar al estudiante para conocer más sobre su el entorno en el que se desarrolla buscando descubrir situaciones de riesgo para un alumno que recién inicia la carrera universitaria. La tutoría de trayectoria refiere a los semestres intermedios de la carrera, podríamos definirlos entre cuarto y séptimo semestre aproximadamente, el objetivo de la tutoría es continuar con el seguimiento del estudiante analizando sus



Fig.3. Vista de seguimiento del estudiante con actividades y reportes, con opción a análisis de clustering. Elaboración propia.

fortalezas, debilidades, amenazas y riesgos, apoyado en la evaluación académica detectando riesgos en materias posibles casos de artículo. Por último, la tutoría de egreso busca apoyar a los alumnos en la etapa final de su carrera, orientando y brindando el apoyo necesario para concluir de manera exitosa la carrera. El trabajo en conjunto del asesor permitió el establecimiento de formularios que nos brinden datos relevantes sobre los estudiantes en los primeros tipos de tutorías: inicio y trayectoria. Dichas actividades solo se pueden contestar cuando existe una tutoría en curso y se relaciona a cierto tipo, de lo contrario estas se encontrarán bloqueadas. El uso de Javascript permitió que los formularios lograrán brindar una mejor interacción con el usuario, expandiendo las preguntas según la respuesta seleccionada y visualizando definiciones de términos que probablemente ocasionarían confusión. La consulta de tutores y coordinaciones podía significar un apoyo para el usuario estudiante, por lo que el administrador dentro de su panel puede agregar o modificar coordinaciones con fines informativos. Los estudiantes y tutores pueden cambiar de datos en el tiempo que signifique la necesidad de actualización de información, por lo que se desarrolló una vista para ambos usuarios donde puedan modificarla sin problema. Django se caracteriza por la variedad de herramientas que proporciona para el desarrollo web, por lo tanto, en búsqueda de proteger las diferentes vistas del sistema se han utilizado los decoradores, estos restringen el acceso según ciertas características específicas lo que para nosotros las definen los grupos de usuario.

Podría considerarse hasta este punto como la primera etapa de desarrollo, la cual incluye el diseño de las diferentes vistas y la integración de las funcionalidades ya mencionadas, para ello conforme el proyecto avanzaba se realizaron pruebas unitarias, de integración y usabilidad para garantizar su correcto funcionamiento.

La segunda etapa de desarrollo del sistema se puede definir en la creación del chatbot y la visualización de datos mediante la clusterización y estadísticas, es decir la implementación del módulo soft computing. El objetivo principal del chatbot es crear una herramienta con la capacidad de responder dudas o preguntas que tengan los estudiantes, sin embargo, es necesario establecer límites y fijar el alcance, para ello decidimos acotar la información sobre los reglamentos de la Universidad de Guadalajara relevantes para los alumnos como lo son servicio social, titulación, evaluación y promoción. El equipo procedió a dar lectura para rescatar los puntos más importantes para su posterior carga del chatbot. El desarrollo comienza con la investigación del modelo a implementar dado que existen distintas alternativas para su creación finalmente elegimos el modelo de red neuronal dado que es una buena herramienta para el procesamiento del lenguaje natural. Esta herramienta puede aprender patrones complejos y generar respuestas apropiadas basadas en las aportaciones del usuario. El modelo se entrena con un conjunto de datos de intenciones y respuestas, lo que le permite aprender los patrones y generar respuestas apropiadas. El modelo se representa como un modelo de aprendizaje

profundo con una capa de entrada, una capa de incrustación, una capa recurrente y una capa de salida. La primera capa toma la entrada del usuario, la capa de incrustación convierte la entrada en un vector, la capa recurrente procesa el vector y genera una respuesta, y la capa de salida genera una respuesta basada en el vector procesado. Dado que el framework Django trabaja con el lenguaje Python resultó más sencilla la implementación dentro del proyecto, para ello se utilizó la librería de TensorFlow y Keras las cuales nos proporcionan una facilidad y flexibilidad en cuanto al desarrollo de redes neuronales. El desarrollo técnico se basa en diferentes etapas iniciando por la creación de un modelo secuencial, este es apropiado para una pila simple de capas donde cada capa tiene exactamente un tensor de entrada y un tensor de salida. Se agrega una capa densa (totalmente conectada) con 128 neuronas. La función de activación ReLU (Rectified Linear Unit) se utiliza para introducir no linealidad en la red. Continuando tenemos la agregación de una capa de dropout para regularización. El dropout aleatoriamente apaga un porcentaje de las neuronas durante el entrenamiento, lo que ayuda a prevenir el sobreajuste. Después de esto se agrega otra capa densa con 64 neuronas y la función de activación ReLU, para enseguida tener otra capa de dropout. Finalmente, para la capa de salida se tiene con un número de neuronas igual al número de clases de salida. La función de activación softmax se utiliza para obtener una distribución de probabilidad sobre las clases de salida. Se crea un optimizador SGD (Stochastic Gradient Descent) con una tasa de aprendizaje de 0.001, un momento de 0.9 y habilita Nesterov momentum. Una vez realizado lo anterior se compila el modelo con la función de pérdida 'categorical cross entropy', ya que estamos tratando con un problema de clasificación multiclase.

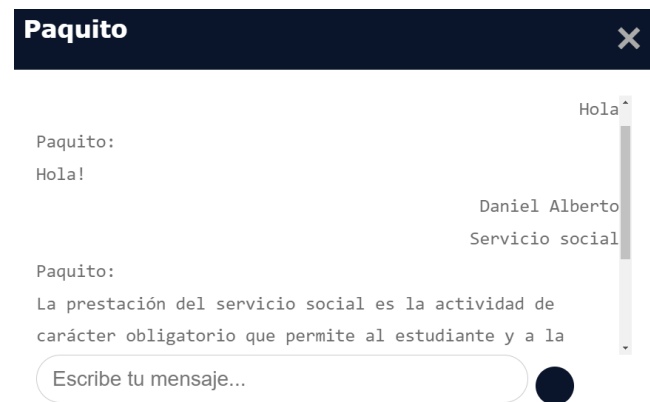


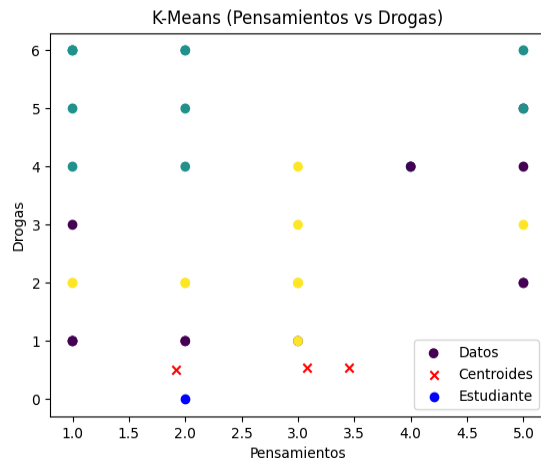
Fig. 4. Ventana de chatbot. Elaboración propia.

El optimizador SGD definido anteriormente se utiliza para minimizar esta función de pérdida. Se define 'accuracy' como la métrica para evaluar el rendimiento del modelo durante el entrenamiento. Seguido a esto se entrena el modelo utilizando los datos de entrenamiento. Se especifica que el entrenamiento se realizará durante 100 épocas con un tamaño de lote de 5. Por último, el modelo entrenado se guarda en un archivo junto con la información de entrenamiento. Esto permite cargar el modelo

entrenado posteriormente para su uso sin necesidad de volver a entrenarlo desde cero.

La implementación del chatbot aprovecha la popularidad de tecnologías como Chat GPT que facilita la consulta de información sin necesidad de la búsqueda manual en internet para su posterior filtrado de fuentes y su lectura, esto puede parecer mucha tarea hoy en día para el usuario, por lo que brindar mayor facilidad para mostrar la información requerida para el estudiante sobre dudas específicas de los reglamentos ya mencionados genera un valor sobre la herramienta.

Complementando el módulo de soft computing decidimos utilizar la información que el sistema es capaz de recopilar derivado de las actividades contestadas por los estudiantes en las tutorías de inicio, las cuales abordan temas importantes como adicciones, empleo, pueblos indígenas, comunidades LGBT, neurodiversidad, salud mental entre otros. La visualización de estos datos puede permitir al administrador o en su caso coordinador tomar decisiones o estrategias para la mejoría de los planes de tutoría. Debido a la dificultad que representa la etiquetación de las respuestas, decidimos agrupar a los usuarios mediante el algoritmo de K-means que no necesita etiquetas y solo calcula la distancia basándose en el número de centroides.



Fuma algún tipo de droga
 0: Ninguno 1: Tabaco 2: Marihuana 3: Cristal 4: Fentanilo 5: Cocaína 6: Otro
Pensamientos sobre quitarte la vida

Fig. 5. Relación entre el tipo de droga consumida y los pensamientos sobre quitarse la vida. Elaboración propia.

Dado que la actividad 1 de tutoría de inicio contiene un gran número de preguntas, optamos por escoger las principales que realmente influyen dentro de un estudiante, es con estos valores que trabaja el algoritmo. La facilidad de la elección de las preguntas se encuentra en los valores almacenados, es decir son numéricos por lo que reduce el trabajo de desarrollo e implementación. Los campos seleccionados de la actividad son los siguientes: ¿Trabaja actualmente?, Horas trabajadas, ¿Con quién vive actualmente?, ¿Recibe algún apoyo económico?, ¿Pertenece a alguna comunidad LGBT?, ¿Cuántos hijos tiene?, ¿Fumas?, ¿Tomas alcohol?, ¿Acudes a terapia?, frecuencia de

tomar alcohol, tipos de droga, dificultades para conciliar el sueño, cansancio o fatiga que te impiden realizar algunas actividades, miedos incontrolables o fobias a objetos, personas o situaciones, irritación o mal humor, ansiedad o angustia generalizada que te impide concentrarte, motivación para hacer tus actividades cotidianas, pensamientos sobre quitarte la vida, experimento tristeza, en general me siento feliz. Cabe mencionar que mediante los modelos de Django (ORM) se manejan tales campos con valores enteros entre distintos rangos, ejemplo 1 a 5.

Mediante la librería de sklearn que proporciona el modelo de k means se le asignan 3 centroides buscando la clusterización en diferentes niveles posibles como bajo, medio y alto refiriéndose al nivel en el que se encuentra el usuario según las evaluaciones. Las respuestas que se evalúan del estudiante son las últimas registradas de actividad trabajando con todo el registro almacenado. Para el modelo de pruebas se probó con al menos 32 cuestionarios contestados. Para su visualización utilizamos la librería de matplotlib, y debido a que son diferentes dimensiones o campos con los que se trabajan, se crearon 4 principales gráficas en las que se compara la relación ejemplo entre la frecuencia de tomar alcohol y la experimentación de tristeza.

El tener una visión amplia sobre las respuestas de los estudiantes dentro de las tutorías de inicio puede permitir la mejora en los programas de tutoría, así como el centralizar el enfoque de esfuerzos dentro del proyectos, es por esto que se implementa un dashboard como herramienta que busca brindar la facilidad de visualización de esta información, generando valor al Big Data que se puede obtener con el sistema. Los datos que se consideraron relevantes a consultar son los siguientes: horas de estudio que los alumnos dedican fuera de la escuela, adicciones, neurodiversidad, comunidad LGBT, tema laboral, frecuencia de sentimiento de tristeza entre los alumnos. La implementación de las gráficas se realiza de forma dinámica, es decir el usuario podrá en algunas gráficas interactuar con el desglose de información. Para ello se buscó crear un diseño que permitiese a cualquier usuario sin tener alto conocimiento en el área de tecnologías de la información para su uso y consulta.

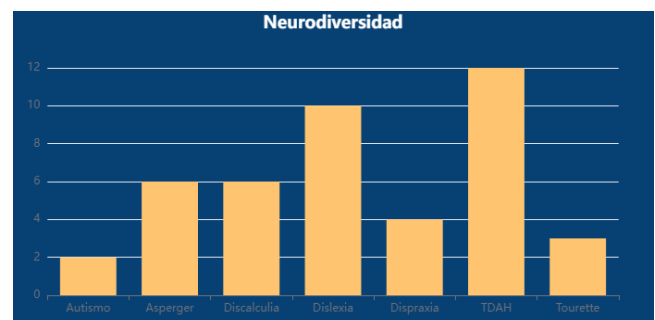


Fig.6. Gráfica que muestra el número de respuestas obtenidas de alumnos que se identifican con alguno de los tipos de neurodiversidad. Elaboración propia.

La última etapa dentro del desarrollo del sistema refiere al despliegue del proyecto, para ello la arquitectura elegida fue cliente-servidor lo que permite que múltiples dispositivos

(clientes) se conecten a un servidor central para acceder a recursos o servicios, brindando posibilidades de escalabilidad, facilidad de mantenimiento, flexibilidad entre otras.

La base de datos hasta el momento se había trabajado de manera local a través del uso de un contenedor Docker utilizando MySQL, por lo que para el despliegue de la base de datos se optó continuar con esta misma forma de trabajar lo que nos llevó a utilizar los servicios de Azure en donde evaluamos el mantener el contenedor otorgando disponibilidad de los datos en todo momento debido a que se encuentra como una instancia de contenedor que se encuentra activo en todo momento, teniéndolo disponible y accesible en todo momento. Para el despliegue se subió la imagen del contenedor de nuestra base de datos al registro de contenedores de Azure para poderla tomar al momento de crear la instancia de contenedor. Se creó la instancia del contenedor usando la imagen previamente creada, asignando los recursos y eligiendo como sistema operativo Linux el cual usa Oracle Linux Server 8.8 que nos otorga ventajas como portabilidad para encapsular todas las dependencias y configuraciones necesarias para ejecutar una aplicación, despliegue rápido para los contenedores pues se pueden crear, iniciar y detener rápidamente. Esto permite implementar y escalar aplicaciones de manera ágil y eficiente.

El aislamiento es para que los contenedores utilicen espacios de nombres y grupos de control del kernel de Linux para proporcionar aislamiento a nivel de sistema operativo. Esto significa que cada contenedor tiene su propio entorno aislado, lo que ayuda a prevenir conflictos entre aplicaciones y mejora la seguridad. En las configuraciones finales de red, se abre el puerto 3306 para realizar la conexión a la base de datos en MYSQL la cual corre en el puerto previamente mencionado.

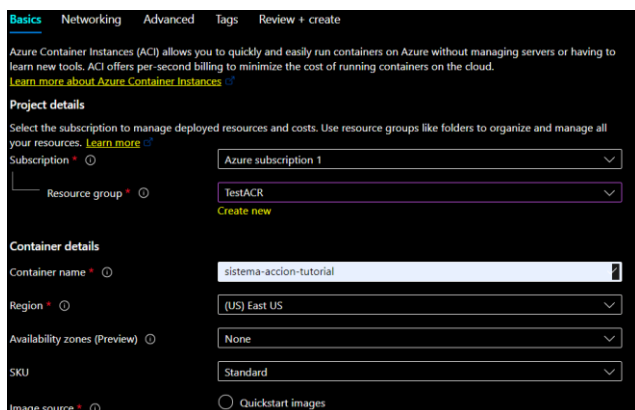


Fig. 7. Creación de instancia en Azure. Elaboración propia.

En el mercado existen variedad de opciones para el despliegue de proyectos desarrollados en Django como lo son Heroku, PythonAnywhere, Google Cloud Platform, Amazon Web Services, Microsoft Azure entre muchos más. Sin embargo, un punto importante de analizar es la factibilidad que proporcione a las dependencias utilizadas dentro del proyecto, para ello el equipo realizó una investigación, así como pruebas donde la opción más viable para el despliegue resultó en AWS (Amazon Web Services).



Fig. 8. Creación de máquina en AWS. Elaboración propia.

Serverless es un modelo de computación en la nube en el cual el proveedor de servicios en la nube es responsable de la infraestructura subyacente, la escalabilidad automática y la gestión de servidores. Esto permite utilizar eficientemente los recursos y representa un ahorro de costos significativo, ya que se basa en el principio de "pagas por lo que consumes". Tanto para el despliegue del proyecto como de la base de datos se utiliza el modelo anteriormente mencionado, significando una ventaja y una aplicación amplia de tecnologías. Trabajamos con Amazon Web Services EC2(Elastic Computer 2) el cual nos permite crear una instancia para alojar una máquina con sistema operativo Linux usando Ubuntu, esto nos otorga diversas ventajas al momento del despliegue del proyecto en DJANGO como el control absoluto del entorno que permite personalizar la configuración de red, seguridad, almacenamiento, instalación y configuración del entorno virtual para instalar todas las dependencias necesarias para el proyecto en Django. Seguridad AWS nos ofrece una amplia gama de herramientas y características de seguridad para proteger la instancia EC2 y datos, incluyendo grupos de seguridad para manejar el acceso a la instancia. En este caso se abrió el puerto 8000 a través del grupo de seguridad para permitir el acceso al proyecto debido a que se ejecuta en el puerto 8000.

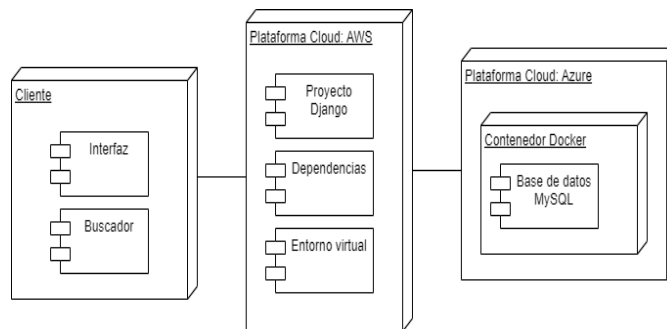


Fig. 9. Diagrama de despliegue del proyecto. Elaboración propia.

D. Módulo I. Gestión de la tecnología de información

El desarrollo del proyecto se constituye en base a la metodología SCRUM con herramientas de apoyo tales como Google Meet, GitHub y diagramas de Gantt para la administración de actividades.

E. Módulo II. Sistemas robustos, paralelos y distribuidos

Implementación de arquitectura cliente-servidor, base de datos en un contenedor Docker local durante la fase de desarrollo, despliegue del proyecto en servicios de nube (AWS) y servicios Azure para la base de datos. Modelo de computación en la nube utilizado como serverless.

F. Módulo III. Cómputo flexible (softcomputing)

Integración de algoritmo k means para la clusterización de estudiantes evaluados en las respuestas de una actividad específica. Desarrollo de un chatbot utilizando redes neuronales para el procesamiento de lenguaje natural basado en patrones y respuestas. Aprovechamiento de la recopilación y almacenamiento de información para brindar una herramienta de Business Intelligence para la toma de decisiones informadas y estrategias en cuanto a los programas de tutoría.

IV. RESULTADOS OBTENIDOS DEL PROYECTO

Los resultados obtenidos después del desarrollo del proyecto son los siguientes: se dio seguimiento y control de solicitudes de y asignación de tutorías, se logró un seguimiento de desempeño de tutores y alumnos mediante la creación de reportes en sistema, se brindó un análisis de información recolectada mediante las actividades contestadas por alumnos mediante estadísticas y clusterización. Chatbot como herramienta que proporciona un apoyo básico respondiendo a dudas del Reglamento de Evaluación y Promoción, Reglamento de Servicio Social y Reglamento de Titulación de la Universidad de Guadalajara.

V. CONCLUSIONES

El éxito del desarrollo de este proyecto se basó en el análisis y comprensión del problema, así como la definición de las necesidades y objetivos a cumplir lo que permitió tener claro el alcance del proyecto. La implementación e integración de distintas tecnologías representan un reflejo de la experiencia y conocimientos adquiridos durante la carrera. Este proyecto impacta en primera instancia a la comunidad universitaria otorgando una herramienta que busca la orientación y comunicación con el estudiante buscando ser una base de apoyo para reducir la deserción escolar del centro universitario brindando una orientación académica y profesional durante el transcurso de la carrera que permita tener mayor cantidad de alumnos egresados para enfrentar los diversos retos del campo laboral profesional. Además, permite el análisis de los datos recolectados mediante actividades de entrevistas contestadas por los alumnos que responden a temas relevantes en el problema de deserción escolar. El crecimiento de esta plataforma no se encuentra limitado por lo que se propone extender la posibilidad de almacenar documentación importante de alumnos tales como diagnósticos médicos, incrementar el conocimiento del chatbot, crear un canal de difusión mediante notificaciones o el aumento del análisis de la información recopilada.

RECONOCIMIENTOS

A nuestro profesor Víctor Manuel Zamora Ramos quien nos orientó y apoyó en el transcurso del desarrollo del proyecto.

A los distintos creadores de contenido y cursos tanto en plataformas como Udemy y Youtube quienes nos otorgaron las herramientas para el desarrollo dentro de este framework.

REFERENCIAS

- [1] Django.DjangoProject.[Online].Available:<https://www.djangoproject.com/>
- [2] AWS | Cloud Computing - Servicios de informática en la nube. Amazon Web Services,Inc.[Online] .Available: <https://aws.amazon.com/es/>
- [3] ¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS.AmazonWebServices,Inc.[Online].Available:<https://aws.amazon.com/es/what-is/scrum/>
- [4] Algoritmo k-means: ¿Qué es y cómo funciona? Thinking for Innovation.[Online].Available:<https://www.iebschool.com/blog/algoritmo-k-means-que-es-y-como-functiona-big-data/>
- [5] ¿Qué es business intelligence? | IBM. IBM in Deutschland, Österreich und der Schweiz. [Online]. Available: <https://www.ibm.com/mx-es/topics/business-intelligence>
- [6] (2002) The IEEE website. [Online]. Available: <http://www.ieee.org/>
- [7] ¿Para qué sirve un dashboard? Software para Dashboards | TuDashboard.[Online].Available:<https://tudashboard.com/para-que-sirve-un-dashboard/>
- [8] Apache ECharts. Apache ECharts. [Online]. Available: <https://echarts.apache.org/en/index.html>
- [9] Factores intra y extra escolares asociados al rezago educativo en comunidades vulnerables. Sistema de Información Científica Redalyc, Red de Revistas Científicas. [Online]. Available: <https://www.redalyc.org/journal/4677/467751868007/html/>
- [10] H.Costa Guzmán. (2024, Marzo). Curso Practico Django. [Online]. Available:https://www.udemy.com/share/101Ys83@F_dganPNrUpxtZw4Ln0tVXrMDWZlvE5HcLUiV58x-8QldvBm73P-F3CdlsKLzrvfuw
- [11] AWS re:Invent 2023 - Compute Innovation Talk. (s. f.). [Vídeo]. Amazon Web Services, Inc. <https://aws.amazon.com/es/ec2/>
- [12] Módulo 2: Crear una instancia EC2. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/getting-started/hands-on/deploy-wordpress-with-amazon-rds/3/>