



Aplicación multiplataforma de modelo computacional predictivo

Cross-platform application of predictive computational model

Autores

✉ **Jéssica Johanna Morales Carrillo*



✉ *Luis Cristóbal Cedeño Valarezo*



✉ *Victor Joel Pinargote Bravo*



✉ *Jesús Stefano Cajape Bravo*



✉ *Jonathan Geovanny Ormaza Calderón*



Grupo de Investigación SISCOR,
Carrera de Computación, Escuela
Superior Politécnica Agropecuaria de
Manabí Manuel Félix López. El Limón
vía a Calceta - El Morro, Ecuador.

* Autor para correspondencia

Comó citar el artículo:

Morales Carrillo, J.J., Cedeño Valarezo,
L.C., Pinargote Bravo, V.J., Cajape
Bravo, J.E. & Ormaza Calderón, J.G.
(2023). Aplicación multiplataforma
de modelo computacional predictivo.
*Informática y Sistemas: Revista
de Tecnologías de la Informática y
las Comunicaciones*, 7(2), 45-51.
<https://doi.org/10.33936/isrtic.v7i2.5832>

Enviado: 30/03/2023

Aceptado: 01/06/2023

Publicado: 03/07/2023

Resumen

El objetivo de esta investigación fue realizar una aplicación que permita la integración de un modelo computacional a una aplicación multiplataforma, para ello se tuvo como datos, la información de un modelo de clasificación de aprendizaje supervisado respecto al levantamiento de información sobre la deserción estudiantil en el contexto del covid-19, esta investigación fue desarrollada en la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López. La metodología que se planteó para el respectivo desarrollo fue Extreme programming (XP). Partiendo de un modelo entrenado, para lo cual se utilizó el framework flask basado en python, luego se procedió a crear un servicio para consumir el modelo, posteriormente se realizaron las respectivas pruebas utilizando postman como cliente web. Se obtuvo como principal resultado dentro de la investigación, que el Framework Flask permitió desarrollar e implementar un API RES de forma mucho más ágil y fácil para el consumo de datos del modelo predictivo.

Palabras clave: Modelo Computacional, flask, modelo como servicio, modelo API.

Abstract

The objective of this research was to carry out an application that allows the integration of a computational model to a multiplatform application, for which data was taken, the information from a supervised learning classification model regarding the collection of information on student dropout in the context of the covid-19, this research was developed at the Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López. The methodology that was proposed for the respective development was Extreme programming (XP). Starting from a trained model, for which the python-based flask framework was used, then we proceeded to create a service to consume the model, later the respective tests were carried out using postman as a web client. The main result of the research was that the Flask Framework allowed the development and implementation of a RES API in a much more agile and easy way for the consumption of data from the predictive model.

Keywords: Computational model, flask, model as a service, API model.



1. Introducción

La evolución de la tecnología ha avanzado tanto hasta el punto de facilitar la vida del ser humano de muchas formas, como es el caso de las redes neuronales convolucionales; si bien es cierto que para utilizarse se debe contar con varios recursos, y algunos de ellos son computacionales y de datos (Bonales-Daimel & Mañas-Viniegra, 2021).

En la actualidad las organizaciones han comprendido la importancia de implementar algún sistema como medio de apoyo a la certificación de contratos, el cual les sirva como soporte para la toma de decisiones en la emisión correspondiente y tener control sobre los datos y la información de diversos documentos que se ejecutan, por lo que permite mejora en disponibilidad y confiabilidad, siendo estas organizaciones tales como: Microsoft azure, Amazon web, IBM, Web Service (Ghimire, 2020). Así mismo, la aplicación de técnicas para inteligencia de datos se vuelve cada vez más común en diversas áreas del conocimiento, desde astronomía (clasificación de exoplanetas) (González, 2021), actividades escolares (Gallegos, et. al., 2020), análisis de sentimientos (Cedeno-Moreno & Vargas, 2020), inteligencia emocional (Romero, et. al., 2022), actividades artísticas como la música (Tepepa, et. al., 2018) entre muchas, que hacen que el procesamiento de datos para obtener patrones con la información permite de alguna manera determinar estrategias o tomar decisiones en base a estos criterios.

En la actualidad, muchas instituciones educativas han emprendido estudios referentes a minería de datos (EDM, Minería de datos educativos), relacionados a tendencias en el aprendizaje o competencias pedagógicas (Díaz, Romero, Faouzi, & Pardo, 2022); así mismo existen estudios referente a la deserción estudiantil en un intento por comprender las causas subyacentes y abordar el problema de manera efectiva. Dos ejemplos notables incluyen la Universidad de Colombia (UNISANGIL) (Sposito et al., 2022) y la Universidad Nacional de La Matanza (UNLaM) (Aparicio et al., 2022). Ambas instituciones han utilizado técnicas de minería de datos para analizar grandes conjuntos de datos y obtener resultados precisos sobre las causas de la deserción estudiantil en sus respectivas carreras. Estos resultados se han utilizado para informar la toma de decisiones y desarrollar estrategias efectivas para prevenir la deserción estudiantil y mejorar la experiencia educativa.

Para aplicar estas tendencias en la minería de datos, es necesario contar con expertos que desarrollen e identifiquen los algoritmos adecuados y efectúen el análisis correspondiente, es por ello que se hace necesario desarrollar aplicaciones que permitan ser consumidores de esta información con interfaces comprensible para cualquier usuario, y que muestren los resultados deseados, de acuerdo con lo requerido.

Las técnicas para consumir modelos computacionales son métodos estandarizados que permiten a los desarrolladores de software integrar modelos computacionales en sus aplicaciones. Las técnicas para consumir modelos computacionales se han establecido para facilitar el acceso a estos datos por parte de los desarrolladores de aplicaciones, y se han diseñado para ser compatibles con la forma en que los desarrolladores de aplicaciones trabajan (Zavala, 2020).

Dentro del esquema de trabajo más utilizado para consumir modelos computacionales está Flask, la cual es un microframework de Python que se basa en el principio de la inversión de dependencias; de acuerdo con Ghimire (2020), Flask brinda simplicidad, flexibilidad, control detallado y un aprendizaje rápido y fácil. Su objetivo es simplificar el desarrollo de aplicaciones web. Flask provee una forma de iniciar la aplicación y hace fácil el proceso de enrutamiento y manipulación de request/response (Ghimire, 2020). El autor Relan (2019) en su libro “Creación de API REST con Flask”, describe que el trabajar con Flask permite desarrollar, implementar y administrar API REST con código Python fácil de leer y comprender.

El uso de aplicaciones con minería de datos en el campo de la educación es, por supuesto, de gran importancia para resolver el problema de la deserción estudiantil. Primero, la deserción es un problema importante en muchas instituciones educativas y puede tener consecuencias significativas para los estudiantes, las instituciones y la sociedad en su conjunto; la identificación temprana de los factores que contribuyen a la deserción estudiantil puede permitir a las instituciones tomar medidas para prevenirla y de alguna manera asegurar el éxito académico de los estudiantes.

En segundo lugar, la extracción de datos puede proporcionar una comprensión más profunda y completa de los factores que influyen en la deserción de los estudiantes, lo que puede ser difícil de determinar con los métodos de investigación tradicionales. Mediante el análisis de grandes conjuntos de datos, la minería de datos puede revelar patrones y tendencias que conducen a la deserción de los estudiantes y ayudar a las instituciones a desarrollar intervenciones específicas y personalizadas para abordar estos problemas.

En tercer lugar, el uso de la minería de datos en la educación no solo puede ayudar a prevenir la deserción de los estudiantes, sino también a mejorar la experiencia educativa en general. Al analizar los datos de los estudiantes, las instituciones pueden identificar patrones de aprendizaje y adaptar la instrucción para satisfacer las necesidades individuales de los estudiantes, mejorando la retención estudiantil y el rendimiento académico.

2. Materiales y Métodos

El objetivo principal del presente trabajo es desarrollar una aplicación para consumir modelo predictivo en caso de abandono de estudio aplicado en la ESPAM MFL. Para lograr esta meta, se estableció como primera actividad consumir un modelo computacional como servicio y posteriormente crear una Aplicación para mostrar los resultados obtenidos. Dentro de esta actividad, se utilizó la metodología XP (eXtreme Programming), siendo esta un enfoque ágil y flexible de desarrollo de software que se centra en la colaboración, la comunicación y la entrega continua de valor (Jiménez Builes et al., 2019). De acuerdo con la definición de la metodología y sus fases, se desarrolló el trabajo tal como se describe a continuación:

Planificación: En esta fase, se realizaron reuniones periódicas con los interesados, que incluyeron miembros del equipo de investigación y otros actores relevantes (Integrantes del proyecto de investigación titulado “Minería de datos para predecir la continuidad de estudios en la educación superior: Caso COVID-19”) con el objetivo de identificar las características principales del producto. Se trabajó en dos equipos de trabajo, donde un equipo se encargó del desarrollo del modelo predictivo, y el otro, el desarrollo de la aplicación para consumir esos datos. De acuerdo con esto, el desarrollo del presente artículo se enfoca exclusivamente a la presentación de los resultados del desarrollo de la aplicación. Las reuniones se llevaron a cabo con bastante frecuencia y se discutieron diversos temas, como los requisitos de funcionalidad, la selección de las herramientas de desarrollo y los posibles riesgos o desafíos del proyecto. Para la creación del producto, se definió el uso de herramientas como el Framework Laravel, que se eligió por su flexibilidad y escalabilidad en el desarrollo de aplicaciones web en PHP, y el framework Flask, que se seleccionó por su capacidad para crear servicios web RESTful en Python. Además, se estableció un plazo de entrega para el producto final. Se identificaron posibles riesgos y desafíos en esta fase, como la disponibilidad de datos de calidad y la integración de diferentes tecnologías.

Diseño: Se utilizó una arquitectura de tres capas, diseñada para el consumo de modelos computacionales. La capa de datos consume el modelo computacional, mientras que la capa de negocios o de servicios actúa como intermediaria entre el cliente y el modelo computacional. La capa de presentación se encarga de mostrar los resultados del modelo al usuario final.

Para asegurar una comunicación eficiente entre las diferentes capas, se definieron interfaces de programación de aplicaciones (APIs) para la transferencia de datos. El modelo computacional utilizado en el proyecto se seleccionó cuidadosamente y se integró en la arquitectura de tres capas. Además, se elaboró el diseño principal de la interfaz gráfica con su respectivo prototipo para ubicar funcionalidades estáticas y así lograr tener una visión general del producto final.

Codificación: El desarrollo del proyecto se ejecutó entre dos desarrolladores, utilizando el lenguaje de programación Python y los frameworks Laravel y Flask para la creación de los servicios. Para mantener un alto nivel de calidad en el código fuente, se puso un gran énfasis en la legibilidad del código. Se llevó a cabo una

consulta continua con otros miembros del equipo involucrados en el proyecto para obtener retroalimentación sobre la claridad y la facilidad de lectura del código, lo que ayudó a identificar y corregir cualquier problema de legibilidad.

El control de versiones del código fuente se gestionó a través de Git, utilizando una rama principal para el desarrollo y ramas secundarias para las correcciones y mejoras. Se utilizaron herramientas de automatización de pruebas, como Pytest, para garantizar la calidad del código y se implementó un proceso de revisión de código por pares para validar la calidad de los cambios antes de ser integrados a la rama principal. La colaboración entre los dos desarrolladores se gestionó mediante herramientas de comunicación en línea, como Discord y Meet. Las tareas se dividieron en pequeñas unidades de trabajo y se asignaron de acuerdo con las habilidades y experiencia de cada desarrollador. Para gestionar las dependencias entre las tareas, se utilizaron técnicas de integración continua y entregas frecuentes de software.

Para mostrar cómo se utilizó esta arquitectura mencionada anteriormente en la práctica, se desarrollaron distintos módulos, el más representativo es el de predicción de continuidad de estudios en la educación superior durante la pandemia de COVID-19. En este módulo, la capa de presentación se utilizó para mostrar los resultados del modelo al usuario final en una interfaz web interactiva. La capa de negocios o de servicios manejó las solicitudes de los clientes y procesó los datos para su consumo en el modelo computacional. La capa de datos consumió el modelo computacional y devolvió los resultados a la capa de negocios o de servicios para la comunicación de la información en la capa de presentación.

Pruebas: Se llevó a cabo una evaluación exhaustiva del servicio desarrollado, con el objetivo de verificar su correcto funcionamiento y estabilidad. Para ello el proceso de pruebas se llevó a cabo en múltiples etapas, comenzando con pruebas unitarias para cada uno de los componentes del servicio. Estas pruebas se realizaron utilizando Pytest y se enfocaron en evaluar la funcionalidad y la calidad del código. Posteriormente, se realizaron pruebas de integración para evaluar la interoperabilidad entre los diferentes componentes del servicio. Estas pruebas se realizaron utilizando Postman, que permitió simular las llamadas a los servicios y verificar que los datos se estaban transmitiendo correctamente entre los componentes.

Por último, se llevaron a cabo pruebas de aceptación con los usuarios finales del servicio, con el objetivo de validar que el servicio cumplía con los requisitos establecidos. Se recopilaron comentarios y sugerencias de los usuarios finales para mejorar la usabilidad y la experiencia del servicio.

3. Resultados y Discusión

El objetivo principal del desarrollo de la aplicación, denominada PREDICTOR, fue crear una herramienta capaz de utilizar un modelo predictivo para determinar el riesgo de abandono de estudios en dicentes universitarios. Para lograrlo, se empleó una arquitectura de tres capas específicamente diseñada para





el consumo de modelos computacionales. A diferencia de las arquitecturas de tres capas tradicionales que trabajan con una base de datos, esta arquitectura consume un modelo computacional como objeto en la capa de datos (Figura 1).

La capa de negocios (Figura 2), también conocida como capa de servicios, actúa como intermediaria entre el cliente y el modelo computacional. Aquí se implementan los servicios web que son consumidos desde diferentes plataformas, como aplicaciones web y móviles. Por otro lado, la capa de presentación (Figuras 3 a 8) se mantiene igual que en el modelo tradicional de tres capas, encargándose de la interfaz de usuario, la interacción con el usuario y la presentación de los resultados obtenidos a través del modelo computacional.

En la codificación se utilizaron diversas técnicas y buenas

```

72 @app.route('/retirados', methods=['GET'])
73 def retirados():
74     data = pd.read_csv('dataset/datasetRebuild.csv')
75     result = data.continua.value_counts().to_dict()
76     return result
77     return None
78
79 @app.route('/retiradosCarreras', methods=['GET'])
80 def retiradosCarreras():
81     data = pd.read_csv('dataset/datasetReBuild.csv')
82     result = data.continua==0].carrera.value_counts().to_dict()
83     return result
84     return None
85
86 @app.route('/calificaciones', methods=['GET'])
87 def calificaciones():
88     data = pd.read_csv('dataset/datasetReBuild.csv')
89     result = {"label": "RETIRADOS", "CONTINUA": "value": [data.continua==0].promedio.mean(), data.continua=
90     return result
91     return None
92
93 @app.route('/computador', methods=['GET'])
94 def computador():
95     compu = pickle.load(open('dataset/myDictionaryCompu.pkl', 'rb'))
96     return compu
97     return None
98
99 @app.route('/electricidad', methods=['GET'])
100 def electricidad():
101     elect = pickle.load(open('dataset/myDictionaryElect.pkl', 'rb'))

```

Figura 1. Servicio implementados en Flask.

prácticas de programación, tales como la documentación del código, el uso de nombres descriptivos para variables y funciones, y la realización de pruebas unitarias. Estas prácticas garantizaron una mayor comprensión del código a largo plazo y una mejor calidad del software en general.

Además de las técnicas y buenas prácticas de programación mencionadas, se implementaron métodos para consumir los servicios creados en Flask (Figura 1). Estos métodos permitieron que la interfaz de usuario interactuara con el modelo predictivo de manera efectiva y eficiente. Los servicios web creados en Flask se diseñaron para ser altamente escalables y resistentes a fallos, lo que garantizó que la aplicación PREDICTOR pudiera manejar grandes volúmenes de datos y ofrecer resultados precisos y rápidos. En general, la combinación de estas prácticas de programación y el uso de Flask resultó en una aplicación robusta y confiable.

```

public function postSolicitud(Request $request){
    $response = Http::get('http://127.0.0.1:5000/predecir', [
        'apetito' => $_GET['apetito'],
        'ayuda' => $_GET['ayuda'],
        'computador' => $_GET['computador'],
        'mb' => $_GET['mb'],
        'trabajo_madre' => $_GET['trabajo_madre'],
        'trabajo_padre' => $_GET['trabajo_padre'],
        'síntomas' => $_GET['síntomas'],
        'triste' => $_GET['triste'],
        'virtual' => $_GET['virtual'],
        'count_actividad' => $_GET['count_actividad'],
        'count_familia' => $_GET['count_familia'],
        'count_disp' => $_GET['count_disp'],
        'electricidad' => $_GET['electricidad'],
        'alumbrado' => $_GET['alumbrado'],
        'vias' => $_GET['vias'],
        'carrera' => $_GET['carrera'],
        'promedio' => $_GET['promedio'],
    ]);

    $data=$response->json();
    return view('covid.reporte', compact('data'));
}

```

Figura 2. Código para consumir los servicios

Tras el proceso de desarrollo, se logró crear una interfaz de usuario intuitiva y fácil de utilizar, tal como se describe a continuación:

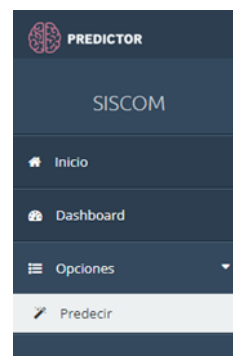


Figura 3. Barra lateral

La interfaz está dividida en tres secciones principales: Inicio, Dashboard y Opciones predecir (Figura 3). La sección de Inicio (Figura 4) es la primera pantalla que el usuario ve al acceder a la aplicación, y ofrece una breve descripción de lo que la aplicación puede hacer, en este caso muestra diferentes tablas con resultados de las predicciones del modelo, tales como alumnos retirados por carrera, total de alumno retirados, promedio de continuidad académica, etc., así como un botón de acceso a las secciones.

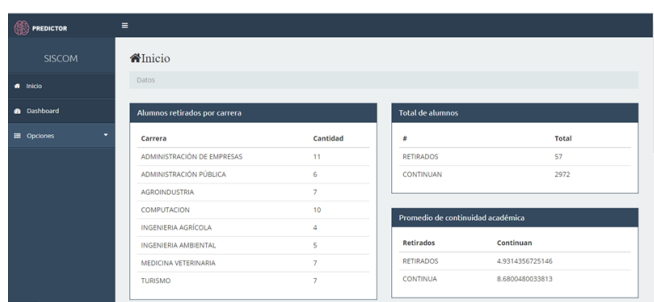


Figura 4. Sección de Inicio con las tablas de los resultados.

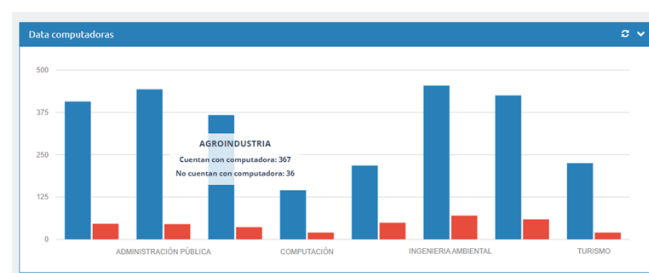


Figura 6. Sección de Dashboard con los gráficos de pastel

La sección de Dashboard (Figuras 5 y 6) es una de las principales características de la aplicación PREDICTOR, ya que es en ella donde se presenta de manera visual los resultados obtenidos por el modelo predictivo.

El objetivo de esta sección es brindar una representación gráfica de los datos para facilitar su interpretación y análisis. En la primera sección, se presenta un gráfico de pastel que indica la cantidad de alumnos que continúan con sus estudios y los que los abandonan en base a los datos obtenidos por el modelo predictivo. Esta figura (Figura 5) permite ver de manera clara la proporción de estudiantes que tienen un riesgo elevado de abandonar sus estudios. En una segunda sección, se presenta un gráfico de pastel que indica la cantidad de alumnos por Carrera que continúan con sus estudios y los que los abandonan en base a los datos obtenidos por el modelo predictivo.

También se identifica una gráfica de barras (Figura 6) que indica la relación entre el acceso a un ordenador en el hogar y el riesgo de abandono de estudios. Este gráfico permite ver si existe una relación entre esta variable y la variable objetivo de la aplicación, lo que puede ser de gran ayuda para comprender mejor el problema del abandono de estudios. Además de estos gráficos, se ofrece la posibilidad de filtrar los resultados por diferentes criterios, como carrera, semestre, entre otros, lo que permite analizar los datos de manera más detallada y obtener conclusiones más precisas.

La sección de Opciones-Predecir (Figuras 7 y 8) es donde se procede a llenar un formulario con preguntas que el modelo necesita para poder predecir y dar resultado, eso implica que se pueda agregar nuevos registros para que la aplicación siempre este actualizada con información reciente, esto con el fin de ayudar para una toma de decisión a las autoridades de la ESPAM MFL.

Figura 7. Sección de Predecir con el formulario de preguntas para poder realizar la predicción

Para el proceso de evaluación de la aplicación, se empleó Postman 7.0.9v., una plataforma popular para probar servicios web. Además, se utilizaron otras herramientas y técnicas de prueba, como pruebas unitarias y de integración, para garantizar que el servicio cumpla con los requisitos funcionales y no funcionales especificados.

Las herramientas de apoyo para la toma de decisiones sean estas de análisis predictivo (machine learning) o descubrimiento de patrones (data mining) se han vuelto muy populares en las diversas áreas de conocimiento, sin embargo, aún es mucho lo que se puede seguir logrando. Es así como el trabajo titulado "Modelo computacional de apoyo al proceso de ingreso a la educación superior cubana" (García, Galarza, & Sepúlveda, 2023), presenta la evaluación favorable de un modelo computacional, misma que fue emitida por especialistas en la actividad de ingreso de todo el país y en el desarrollo de sistemas informáticos, permitió corroborar la relevancia, pertinencia y coherencia del mismo, este modelo fue diseñado para apoyar el proceso de ingreso a la educación y permitió la creación de un sistema informático adaptable y evolutivo, capaz de ajustarse a la dinámica de cambios de los requerimientos del proceso de

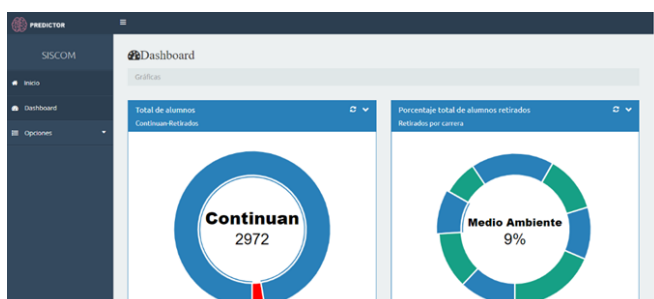


Figura 5. Sección de Dashboard con los gráficos de pastel

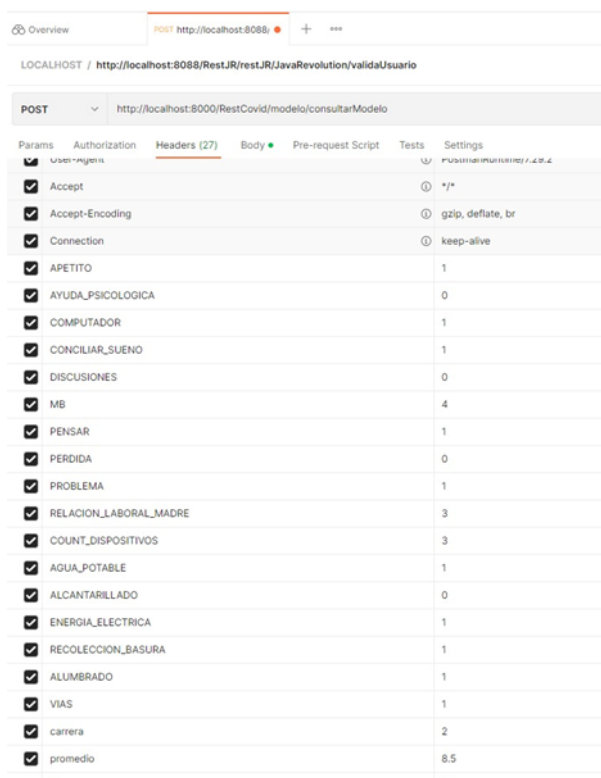


Figura 8. Cliente postman para probar APIs res

ingreso, sin embargo no se pudo identificar las herramientas de apoyo que permitieron su desarrollo. Más bien la investigación se centró en la valoración del modelo y en sustentar la integración de las dimensiones teórica (relaciones esenciales, objetivo, principios, y cualidades) y operacional (núcleos de dominio, diseño e implementación).

Existen en el mercado muchas herramientas de análisis predictivo que abarcan una serie de campos, que incluyen: E-Commerce, marketing de redes sociales, banca y seguros, cuidado de la salud, fabricación, entre otros, y así mismo, tal como lo expresa HashDork (2023) se pueden identificar herramientas gratuitas para este fin, identificando ocho principales tales como: Ciencia, Octava GNU, Mahout Apache, Weka, Scikit-Aprender, Entorno de Software R, Anaconda, Minería de Datos Naranja; donde en la mayoría de los casos se emplean comandos de Python, para lo que se requiere partir de un conocimiento básico en su aplicación. Es por ello que se estima conveniente el trabajar con interfases de aplicaciones que permitan interactuar directamente con el registro de los datos y que estos sean mostrados como

parte de un modelo computacional.

4. Conclusiones

En conclusión, se pudo consumir el modelo computacional como servicio exitosamente, esto permitió crear una aplicación para mostrar los resultados obtenidos del modelo predictivo. Es por ello, que se utilizó la metodología XP (eXtreme Programming) como enfoque ágil y flexible de desarrollo de software, enfocado en la colaboración, la comunicación y la entrega continua de valor.

El uso de herramientas de programación, tales como los frameworks de trabajo, como Laravel y Flask, se ha convertido en una práctica común en el desarrollo de aplicaciones web. Estos frameworks proporcionan un conjunto de herramientas y librerías que facilitan el desarrollo y mantenimiento de proyectos de manera eficiente.

En el caso de implementar un modelo computacional como servicio, el uso de estos frameworks es esencial. Laravel y Flask, son dos de los frameworks más populares en el desarrollo de aplicaciones web, y ofrecen una amplia variedad de funcionalidades para el desarrollo de aplicaciones. Con estas herramientas, se puede implementar un modelo computacional como servicio de forma rápida, y manteniendo un bajo presupuesto. Además, estos frameworks están diseñados para ser escalables y adaptarse a las necesidades de cualquier proyecto, lo que permite expandir el servicio en el futuro si es necesario.

Contribución de los autores

Jessica Johanna Morales Carrillo: Conceptualización, Metodología, Resumen. **Luis Cristóbal Cedeño Valarezo:** Investigación, Análisis de los resultados. **Victor Joel Pinargote Bravo:** Validación de los resultados. **Jesús Stefano Cajape Bravo:** Desarrollo de los resultados. **Jonathan Geovanny Ormaza Calderón:** Ejecución y comprobación de los resultados.

Conflictos de interés

Los autores declaran no tener ningún conflicto de interés.

Referencias bibliográficas

Aparicio, Q., Camilo, J., Chacón, C., & Yamile, L. (2022). Factores de deserción estudiantil School dropout factors. *Revista Matrices Tecnológicas Edición* (14).

- Bonales-Daimel, G., & Mañas-Viniegra, L. (2021). The evolution of advertising investment during a decade of economic crisis (2007-2018). The case of the automotive sector and its adaptation to the Internet. *Revista Mediterranea de Comunicacion* 12(2), 227-243
- Cedeno-Moreno, D., & Vargas, M. (2020). Aprendizaje automático aplicado al análisis de sentimientos. *I+D Tecnológico*, 16(2), 59-66. <https://doi.org/10.33412/idt.v16.2.2833>
- Díaz, E., Romero, M., Faouzi, T., & Pardo, C. (2022). Modelos predictivos de la competencia pedagógica en docentes de EMTP mediante la minería de datos educacionales. *Estudios pedagógicos*, 48(2), 179-197. <https://dx.doi.org/10.4067/S0718-07052022000200179>
- Gallegos, Juan J., Gonzalo Bach1, Luza R. Luna, Bach1, Sulla-Torres José, Gomez-Campos, Rossana and Marco Cossio-Bolaños (2020). Hemispheric Cooperation for Competitiveness and Prosperity on a Knowledge-Based Economy. *LACCEI International Multi-Conference for Engineering, Education, and Technology: "Engineering, Integration, and Alliances for a Sustainable Development"*, 29-31 July 2020, Buenos Aires, Argentina.
- García Ojalvo, I., Galarza López, J., & Sepúlveda, R. (2023). Modelo computacional de apoyo al proceso de ingreso a la educación superior cubana. *Revista Cubana De Educación Superior*, 41(1), 193-208
- Ghimire, D. (2020). *Comparative study on Python web frameworks: Flask and Django*. Metropolia University of Applied Sciences. Bachelor of Engineering.
- González, Johans (2021) *Algoritmos de Aprendizaje Supervisado en la Clasificación de Exoplanetas en Python*. <http://repositorio.uan.edu.co/handle/123456789/5839>. Universidad Antonio Mariño. Facultad de Ingeniería Mecánica, Electrónica y Biomédica.
- HashDork (2023) *Las 8 mejores herramientas de análisis predictivo (código abierto)*. Consultado: 28/03/2023. <https://hashdork.com/es/best-predictive-analysis-tools-open-source/>
- Jiménez Builes, J. A., Ramírez Bedoya, D. L., & Branch Bedoya, J. W. (2019). Metodología de desarrollo de software para plataformas educativas robóticas usando ROS-XP. *Revista Politécnica*, 15(30), 55–69. <https://doi.org/10.33571/rpolitec.v15n30a6>.
- Relan, K. (2019) *Creación de API REST con Flask*. Apress. DOI <https://doi.org/10.1007/978-1-4842-5022-8>.
- Romero, S., Hernández, I., Barrera, R. y Mendoza, A. (2022). Inteligencia emocional y desempeño académico en el área de las matemáticas durante la pandemia. Universidad del Zulia.
- Spositto, O. M., Etcheverry, M. E., Ryckeboer, H. L., & Bossero, J. (2022). Aplicación de técnicas de minería de datos para la evaluación del rendimiento académico y la deserción estudiantil.
- Tepepa, A., Pérez, H., Nakano, M. (2018) Algoritmos de aprendizaje supervisado para la clasificación de géneros musicales caracterizados mediante modelos estadísticos. *Research in Computing Science* 147(5).
- Zavala Diaz, A. G. (2020). Sistema de respaldo de datos automatizado en la nube de amazon web services para evitar la posible pérdida de información en la empresa nessus Hoteles S.A. *Universidad Científica*, 1(1), 1–2.

