

Universidad Internacional de La Rioja (UNIR)

ESIT

Máster Universitario en Inteligencia Artificial

Inteligencia artificial aplicada a la gestión de capacidad de infraestructura tecnológica

Trabajo Fin de Máster

Presentado por: Cevallos Terán, Carlos Xavier

Director/a: García Martínez, Yamila

Ciudad: Quito - Ecuador
Fecha: 23 de septiembre de 2020

Resumen

Un dimensionamiento adecuado de infraestructura tecnológica puede redundar en servicios disponibles y ahorros de costos. Las investigaciones actuales en inteligencia artificial, contemplan proveedores en la nube y otros sectores. No se identifican soluciones para usuarios de infraestructura de centro de cómputo. La contribución de este trabajo es una herramienta inteligente que facilita la gestión de capacidad de infraestructura tecnológica. Se construyó la solución utilizando la metodología CRISP-DM y un caso de aplicación. El sistema contempla la gestión de datos de demanda y capacidad, y la generación de pronósticos con los modelos de aprendizaje supervisado: Regresión Lineal, Random Forest, Máquinas de Vector de Soporte y Redes Neuronales Artificiales. De acuerdo al error cuadrático medio, el mejor modelo fue Random Forest, con un error inferior al 20% respecto al máximo valor de la métrica analizada. De cara al futuro, se plantea el uso y afinamiento de esta herramienta en entornos reales.

Palabras Clave: Gestión de Capacidad, Inteligencia Artificial, Aprendizaje Supervisado

Abstract

A proper dimensioning of the technological infrastructure can result in available services and cost savings. The current artificial intelligence researches consider cloud computing providers and other sectors. None solutions were identified for data center infrastructure users. The contribution of this work is an intelligent tool that makes easier the capacity management of the technological infrastructure. The solution was built using the CRISP-DM methodology and an application case. The system contemplates the management of demand and capacity data, and the generation of forecasts with the supervised learning models: Linear Regression, Random Forest, Support Vector Machines and Artificial Neural Networks. According to the mean square error, the best model was Random Forest, with an error of less than 20% with respect to the maximum value of the analyzed metric. Looking ahead, the use and refinement of this tool in real environments is proposed.

Keywords: Capacity Management, Artificial Intelligence, Supervised Learning

Índice de contenidos

1	Introducción	1
1.1	Retos de la gestión de capacidad tecnológica.....	1
1.2	Propuesta de solución con Inteligencia Artificial.....	4
1.3	Estructura del documento	5
2	Gestión de capacidad y su relación con la IA.....	6
2.1	La gestión de la capacidad tecnológica.....	6
2.2	Aplicación actual de la IA a la gestión de la capacidad tecnológica.....	9
3	Objetivos y metodología de trabajo	13
3.1	Objetivo general.....	13
3.2	Objetivos específicos	13
3.3	Metodología CRISP-DM.....	13
4	Implementación de la solución de IA para la gestión de la capacidad tecnológica	18
4.1	Entendimiento del negocio – caso de aplicación	18
4.1.1	Definición del caso de aplicación – administrador de libros.....	18
4.1.2	Implementación del administrador de libros	22
4.1.3	Definición de requerimientos del sistema inteligente	27
4.2	Entendimiento de los datos	30
4.3	Preparación de datos	36
4.4	Modelamiento de inteligencia artificial	41
4.5	Evaluación – desempeño de los modelos	48
4.6	Interfaces de usuario e instalación de la solución.....	53
5	Conclusiones y trabajo futuro	60
5.1	Conclusiones	60
5.2	Líneas de trabajo futuro	62
6	Bibliografía	63
7	Anexos.....	66

7.1	Anexo I. Código publicado	66
7.2	Anexo II. Artículo de investigación	68

Índice de tablas

Tabla 1. Estado del arte y relación con el presente trabajo	12
Tabla 2. Características de computadores utilizados.....	26
Tabla 3. Ejemplo de error por hora de computadores no sincronizada	35
Tabla 4. Ejemplo de datos de demanda obtenidos desde la Base de Datos	39
<i>Tabla 5. Descripción de la red neuronal implementada</i>	<i>48</i>
Tabla 6. Análisis descriptivo de valores de la métrica “MemoriaUsadaJVM”	49
Tabla 7. Cuadro comparativo de resultados de errores por modelo	51
Tabla 8. Comparación de RMSE de mejor modelo con análisis descriptivo de la métrica “MemoriaUsadaJVM”	52

Índice de figuras

Figura 1. Proceso de gestión de capacidad.....	7
Figura 2. Relación entre la gestión de la demanda y la gestión de capacidad.....	8
Figura 3. Fases de la metodología CRISP-DM.....	14
Figura 4. Patrón de demanda para el caso de aplicación – registro de libros.....	20
Figura 5. Patrón de demanda con factor aleatorio para registro de libros.....	21
Figura 6. Ejemplo de información de libros utilizada para el caso de aplicación.....	23
Figura 7. Arquitectura del caso de aplicación – administrador de libros.....	24
Figura 8. Casos de uso del sistema inteligente de gestión de capacidad.....	28
Figura 9. Arquitectura de componentes de recolección de métricas.....	32
Figura 10. Ejemplo de información generada por el Agente Software Pandora FMS.....	34
Figura 11. Arquitectura de preparación de datos.....	37
Figura 12. Ejemplo de Support Vector Regression con kernel RBF.....	43
Figura 13. Red neuronal por capas y proceso feedforward.....	44
Figura 14. Arquitectura integral de la solución IACT.....	45
Figura 15. Red neuronal implementada para pronósticos de capacidad tecnológica.....	47
Figura 16. Análisis del error cuadrático medio respecto a profundidad de Random forest....	50
Figura 17. Análisis del error cuadrático medio por época en red neuronal.....	51
Figura 18. Tecnologías utilizadas para la implementación de la interfaz de usuario.....	53
Figura 19. Estructura de pantallas del sistema IACT.....	54
Figura 20. Pantalla para gestión de casos del sistema IACT.....	55
Figura 21. Pantallas para gestión de datos de demanda.....	55
Figura 22. Pantallas para gestión de datos de métricas de infraestructura.....	56
Figura 23. Pantallas para la generación del modelo inteligente.....	57
Figura 24. Modelo generado abierto en Jupyter Notebook.....	58

1 Introducción

La infraestructura de hardware habilita los servicios tecnológicos en todo tipo de organizaciones sean estas públicas o privadas, y por consiguiente debe estar bien dimensionada. Entre el 2015 y 2019, de acuerdo a la multinacional Accenture, el 95% de las empresas líderes y el 64% de las empresas rezagadas, han incrementado su presupuesto destinado innovación tecnológica. Adicionalmente, Mercedes Oblanca, de Accenture Technology señala que aquellas organizaciones que sean capaces de maximizar sus inversiones en tecnología tendrán el doble de ingresos (Expansión, 2019). Por otro lado, en el sector gubernamental, de acuerdo a la encuesta realizada en el 2020 por el Departamento de Economía y Justicia Social de las Naciones Unidas se identifica desde el 2018 al 2020 un crecimiento aproximado del 9% en el índice global de desarrollo de gobierno electrónico EGDI (por sus siglas en inglés E-Government Development Index) y en algunos casos un crecimiento cercano al 30% (Naciones Unidas, 2020, p. XXIV-XXV). Bajo este contexto de constante innovación tecnológica tanto en el sector público como privado, y conscientes de que toda solución tecnológica es soportada por hardware (almacenamiento, memoria, procesamiento), es necesario contar con soluciones que permitan determinar de mejor manera los recursos necesarios para optimizar y sacar el mayor provecho de las inversiones que se realicen en infraestructura tecnológica.

1.1 Retos de la gestión de capacidad tecnológica

La gestión de la capacidad es un elemento clave dentro de una organización que depende de la tecnología, ya que permite asegurar que se cuente de forma oportuna con la capacidad de las tecnologías de la información a un costo justificado para cubrir las necesidades actuales y futuras de negocio (Office of Government Commerce, 2007a). Dekkers (2003) afirma que es importante que la gestión de la capacidad se realice a nivel estratégico y que se cuente con una adecuada implementación operativa y táctica.

La gestión de capacidad abarca diferentes actividades que van desde la revisión de la capacidad y desempeño actual hasta el análisis de la demanda transaccional y la estimación de crecimientos y necesidades de recursos de infraestructura, como son: el almacenamiento, el procesamiento, la memoria y el ancho de banda. Y son justamente estas últimas tareas, las que presentan mayor complejidad y dificultad para los administradores de infraestructura; es así que, la estimación de los requisitos de capacidad es considerada como un aspecto avanzado de acuerdo al *“Modelo para el desarrollo de sistemas de servicio de tecnología*

de la información” (MDSSTI-SaaS) mismo que está basado en mejores prácticas de la industria, (Gómez Barroso & Valcarcel Martínez, 2019).

Entre las principales causas de la dificultad y complejidad de realizar las estimaciones y proyecciones de gestión de la capacidad se tiene las siguientes:

- Para realizar el análisis no solo se utiliza la información y conocimiento técnico, sino que también se requiere entender el comportamiento de negocio. Savvides Andy (2004) afirma que la gestión de capacidad debe ser visto como un proceso que considera tanto las necesidades técnicas como las de negocio. Savvides también indica que, este proceso debe iniciar revisando las necesidades actuales de la organización y anticiparse a sus requerimientos futuros.
- En la misma línea del punto anterior, es complejo establecer la relación que existe entre las transacciones de negocio¹ y el uso de recursos tecnológicos por varios factores:
 - *No siempre el consumo es lineal:* no necesariamente el ejecutar 100 transacciones consume 100 veces más lo que consume una. En los sistemas e infraestructura existen varias estrategias de optimización de recursos, así por ejemplo se puede mencionar el uso de un pool de conexiones a bases de datos, en donde se crean un número x de conexiones, independientemente de si se ejecuta 1, 10 o más transacciones que se conecten a la base de datos.
 - *El uso de recursos depende de la concurrencia con la que se realizan las transacciones:* mientras una simple transacción ejecutada con cierta periodicidad, por ejemplo, cada minuto, podría tener un consumo prácticamente imperceptible para el monitoreo de la infraestructura; la misma transacción ejecutada cientos o miles de veces de forma simultánea, podría consumir una parte considerable o la totalidad de los recursos existentes.

Desde el punto de vista tecnológico se contemplan dos tipos de paralelismo el real y el lógico, razón por la cual el número de procesos simultáneos percibidos no se limita únicamente al número de núcleos de un procesador. Adicionalmente, se explica que la programación concurrente tiene entre sus ventajas el uso eficiente de recursos; sin embargo, uno de sus inconvenientes

¹ El término transacción de negocio se utiliza para referirse a actividades a nivel funcional realizadas en los sistemas, por ejemplo: en un banco una transacción de negocio podría ser la creación de una nueva cuenta de ahorros o el realizar una transferencia de valores de una cuenta a otra.

es que pueden obtenerse resultados diferentes en ejecuciones distintas aun cuando se tengan las mismas entradas (Muñoz et al., 2013, pp. 3-5). Esto ocasiona también que pueda existir un consumo distinto de recursos.

- En una organización no se llevan a cabo solo un tipo de transacciones de negocio, sino que se realizan varias transacciones de diferentes tipos y de forma simultánea. Así, por ejemplo, en un sistema contable no solo se crean cuentas, sino que también se registran transacciones contables y se obtienen reportes. Cada tipo de transacción puede hacer un uso diferente de recursos.

En resumen, el relacionar las transacciones de negocio con el consumo de recursos de infraestructura, no puede verse como un simple proceso de medir una transacción y pensar que se obtendrá el consumo de recursos de varias transacciones multiplicando por el total de transacciones.

- Otro inconveniente es la disponibilidad de información: para realizar estimaciones y predicciones, es necesario contar con datos históricos y actuales. Si una empresa no cuenta con la misma será imposible pensar en realizar estimaciones o proyecciones adecuadas.
- No se disponen de soluciones automatizadas que permitan estimar de una manera fácil los requerimientos futuros de capacidad de infraestructura. Los análisis de mediano y largo plazo o no se realizan de manera técnica o se efectúan de forma manual con mucha dificultad.

Los sistemas de administración de infraestructura se enfocan principalmente en la gestión operativa, es decir en el monitoreo de lo que sucede a la fecha y no han incorporado esquemas inteligentes de predicción de requerimientos de capacidad.

Existen dos situaciones asociadas a una mala predicción de la capacidad de la infraestructura tecnológica (Office of Government Commerce, 2007c, p. 129):

- La primera es el sobredimensionamiento de la capacidad, en este caso se adquiere infraestructura en exceso lo que tiene como consecuencia la generación de costos innecesarios.
- La segunda se da cuando la capacidad prevista resulta insuficiente, en cuyo caso el impacto es la calidad del servicio y su crecimiento futuro. La falta de infraestructura generaría lentitud y problemas de disponibilidad de los servicios tecnológicos.

Savvides Andy (2004) explica que la gestión de capacidad no se puede ajustar hacia atrás, si ya se presentó el problema por una pobre capacidad, es muy tarde.

Al momento, no existen soluciones tecnológicas que faciliten una adecuada estimación de recursos tecnológicos de una forma automática e inteligente. El aporte de la investigación se centra en la aplicación de técnicas de Inteligencia Artificial para la identificación de la capacidad de la infraestructura tecnológica dada una demanda transaccional.

1.2 Propuesta de solución con Inteligencia Artificial

El contar con soluciones automatizadas que utilicen herramientas de Inteligencia Artificial permitirá mejorar la calidad de las estimaciones y que estas se realicen de una forma más fácil y estandarizada. La Inteligencia Artificial, a través de las herramientas de aprendizaje automático, harán posible el establecer predicciones para el uso de recursos de infraestructura identificando su relación con la demanda transaccional.

En el presente proyecto se plantea la implementación de un software que tome información histórica de uso de recursos tecnológicos y demanda transaccional, e identifique automáticamente una proyección de necesidades de capacidad de infraestructura utilizando algoritmos de aprendizaje supervisado.

Para la implementación de la solución propuesta se utilizará la metodología CRISP-DM, dentro de la cual se contempla el entendimiento del negocio, el entendimiento y preparación de los datos, la implementación del modelo, su evaluación y la implementación de interfaces para el usuario. Dentro del modelo se realizará el análisis y uso de algoritmos de regresión y sus resultados serán evaluados mediante métricas de calidad como el error cuadrático medio, su raíz cuadrada y el error absoluto medio.

El objetivo principal es proveer una solución automatizada que permita efectuar proyecciones de requerimientos de infraestructura en el uso de la infraestructura para que se realice una adecuada gestión de capacidad dentro de una organización.

Como conclusión del trabajo se tuvo que, entre los modelos implementados, aquel que generó un menor error para los pronósticos de la métrica analizada, "MemoriaUsadaJVM", fue Random Forest. Al comparar la raíz del error cuadrático medio con el mayor valor que puede tener la métrica la proporción fue menor al 20%, considerándose un resultado favorable.

1.3 Estructura del documento

El contenido del presente trabajo se encuentra organizado en los siguientes capítulos:

- **Gestión de capacidad y su relación con la IA:** en esta sección se brinda el marco teórico correspondiente a la gestión de la capacidad de los servicios tecnológicos y su relación con el proceso de gestión de la demanda. A continuación, se describen los principales trabajos identificados en relación al uso de Inteligencia Artificial en el ámbito de la gestión de la capacidad de los servicios tecnológicos.
- **Objetivos y metodología de trabajo:** en este capítulo se detalla el objetivo general, objetivos específicos y la metodología CRISP-DM utilizada para la implementación de la solución.
- **Implementación de la solución de IA para la gestión de la capacidad tecnológica:** en este apartado se inicia con la descripción del caso de aplicación del análisis de capacidad, el cual corresponde a un sistema web de administración de libros. A partir de este caso se definen los requerimientos del sistema inteligente. A continuación, se explica el resultado de cada etapa ejecutada conforme a la metodología y se describe la solución implementada incluyendo la evaluación de los resultados obtenidos por los algoritmos de Inteligencia Artificial implementados.
- **Conclusiones y trabajo futuro:** los elementos más relevantes del trabajo se resumen en este apartado y se plantean propuestas para futuras iniciativas a partir de los resultados alcanzados.

2 Gestión de capacidad y su relación con la IA

En esta sección se introducen los principales conceptos en torno a la gestión de la capacidad tecnológica y su relación con la gestión de la demanda (necesidades de negocio). Así también, se realiza un recorrido de las principales investigaciones que se han realizado los últimos años sobre la aplicación de la Inteligencia Artificial a la gestión de la capacidad tecnológica. Como se podrá evidenciar es un campo con gran potencial y todavía poco explorado.

2.1 La gestión de la capacidad tecnológica

De acuerdo a lo mencionado por Félix-Sánchez y Calvo-Manzano (2015), la gestión de unidades de tecnología cada vez representa mayores retos, antes era suficiente con gestionar y monitorear los servidores y redes, actualizar los equipos y desarrollar e instalar aplicaciones; sin embargo, hoy en día, se requiere de una gestión efectiva y una operación de tecnología predecible y confiable, incorporando la eficiencia en tiempo y recursos. Este enfoque es precisamente para lo que se creó la gestión de la capacidad de los servicios tecnológicos, misma que ha sido abordada a nivel procedimental desde diferentes modelos y estándares entre los cuales se encuentran las normas ISO/IEC 20000:2005, la Biblioteca de Infraestructura de Tecnologías de la Información (ITIL por sus siglas en inglés) y COBIT.

De acuerdo al marco de referencia de ITIL, el objetivo de la gestión de la capacidad es asegurar la capacidad de TI (Tecnologías de la Información) en todas sus áreas para cumplir, de forma oportuna y a un costo justificable, con las necesidades actuales y futuras de negocio (Office of Government Commerce, 2007a, p. 79).

El nivel de madurez en la implementación de la gestión de la capacidad en una organización puede ser analizado a partir del Modelo de Madurez de la Capacidad (CMMI) el cual es respaldado por el marco de gobierno de tecnologías de la información Cobit2019 (ISACA, 2018). En este modelo se establecen 6 niveles con una valoración entre 0 y 5; como se presenta a continuación (ISACA, 2018):

- Nivel 0: es el más bajo y se caracteriza por la ausencia de capacidades básicas del proceso.
- Nivel 1: se tiene una aplicación incompleta de actividades y no se realizan de forma muy organizada.
- Nivel 2: se cumple con el propósito del proceso mediante actividades básicas pero completas.

- Nivel 3: el proceso está bien definido y se lleva a cabo de una manera más organizada.
- Nivel 4: a más de estar bien definido el proceso, se mide de forma cuantitativa su rendimiento.
- Nivel 5: que es el más alto en donde el proceso se encuentra bien definido, se realizan mediciones que permiten mejorar el desempeño y se cuenta con mejora continua.

Como se puede notar en la descripción anterior en los niveles más altos 4 y 5 es indispensable contar con un proceso bien definido y con información cuantitativa que garantice su rendimiento. El marco de referencia ITIL establece para el proceso de Gestión de la Capacidad cuatro acciones principales y un sistema de información de la gestión de la capacidad, como se puede apreciar en la *Figura 1*.

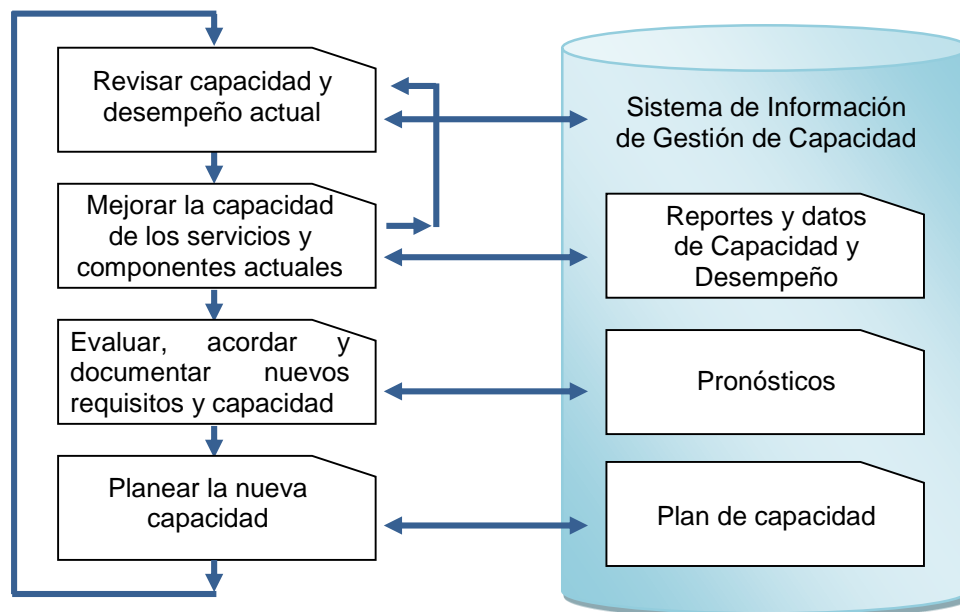


Figura 1. *Proceso de gestión de capacidad.*

Esta figura muestra las actividades principales y el sistema de información que involucra la gestión de la capacidad.

Fuente: Basado en ITIL Service Strategy (*Office of Government Commerce, 2007b, p. 82*)

El presente trabajo brindará una solución para la generación de pronósticos que forma parte del Sistema de Información de Gestión de Capacidad el cual es un insumo de la evaluación de requisitos y capacidad que a su vez alimenta a la planeación de la nueva capacidad. Esto también favorecerá para que la organización fortalezca la información requerida para alcanzar los niveles de madurez 4 y 5.

Para una correcta Gestión de la Capacidad, es importante también comprender el proceso de Gestión de la Demanda, ya que este le provee de información valiosa respecto a los patrones

de actividad del negocio y las necesidades que se deberán satisfacer. La gestión de la demanda de acuerdo a lo que se indica en el libro de Estrategia del Servicio de ITIL es un aspecto crítico y si no es bien manejada puede ser una fuente de riesgo para la provisión de servicios, ya que no se podría advertir la demanda que debe ser soportada por las TI. (Office of Government Commerce, 2007c, p. 129).

Como se puede apreciar en la *Figura 2*, la actividad de negocio tiene una relación directa sobre la provisión de servicios tecnológicos. La carga que deberán soportar dichos servicios depende también de los patrones de demanda que genera la actividad propia de la organización. Esto se convierte en un insumo para el plan de gestión de la capacidad, el cual a su vez se relaciona continuamente con la gestión de la demanda.

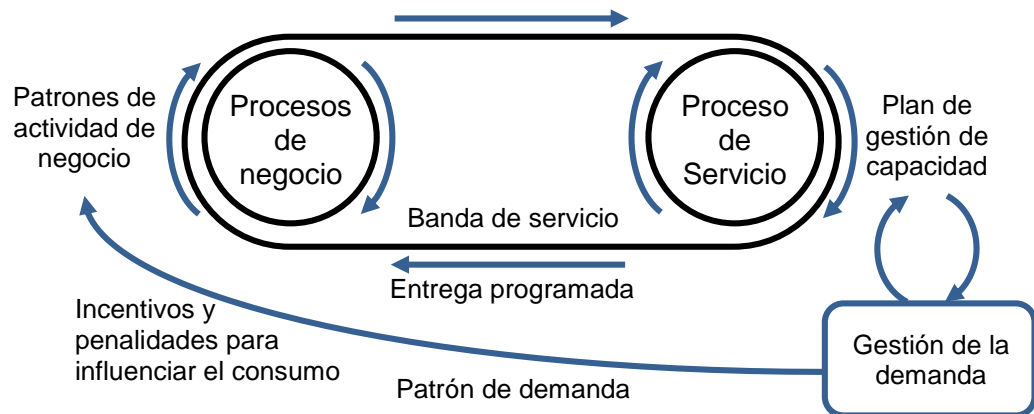


Figura 2. *Relación entre la gestión de la demanda y la gestión de capacidad.*

Esta figura muestra la estrecha interacción que existe entre los procesos de negocio, la gestión de la demanda y la gestión de la capacidad.

Fuente: Basado en ITIL Service Strategy (Office of Government Commerce, 2007c, p. 130)

Es decir, la capacidad requerida de la infraestructura tecnológica de una empresa depende, entre otras cosas, de los patrones de actividad del negocio. Por ejemplo, una empresa de comercio de vestimenta podría requerir mayor capacidad de infraestructura tecnológica en el mes de diciembre considerando que en dicho mes las ventas crecen y se registra mayor cantidad de ventas y facturas.

Bajo este contexto, el presente trabajo busca brindar una solución de software que facilite la gestión de la capacidad de la infraestructura tecnológica a través del uso de la Inteligencia Artificial en las estimaciones de necesidades futuras de infraestructura.

2.2 Aplicación actual de la IA a la gestión de la capacidad tecnológica

Se ha realizado una búsqueda intensiva, en varias fuentes académicas entre las que se destacan: IEEE, Scopus, Springer Journals, Web of science y Proquest Central acerca de trabajos previos de aplicación de la Inteligencia Artificial a la gestión de la capacidad y/o gestión de la demanda asociada a infraestructura tecnológica, entre los trabajos identificados y que tienen relación directa se tiene:

- En los artículos “Intelligent Cloud Capacity Management” de Jiang, Perng, Tao y Rong (2012) y “Cloud Analytics for Capacity Planning and Instant VM Provisioning” (Jiang et al., 2013), los autores describen el uso de técnicas de Inteligencia Artificial para la planeación de capacidad en el contexto de la administración de computación en la nube, más conocida como cloud computing por su nombre en inglés.

El primer artículo se enfoca en la estimación de capacidad y la gestión de un ambiente tipo nube, de manera que en base a las proyecciones de uso futuro se activan o inactivan las máquinas virtuales provistas, se ajustan facilidades como el aire acondicionado del centro de datos y se calendariza el trabajo del personal. Para las estimaciones de capacidad se utilizan y comparan seis esquemas de predicción de línea de tiempo que son: promedio de movimiento, auto-regresión (regresión lineal), redes neuronales artificiales (regresión no lineal), máquinas de vector de soporte (aprendizaje lineal con un kernel no lineal), programación de expresión genética (algoritmo heurístico) y un esquema de ensamblaje de los cinco predictores anteriores. De acuerdo al análisis, el método de ensamblaje es el que tiene mejor desempeño y se confirmó que su uso permitió reducir efectivamente los costos del entorno de nube.

El segundo artículo hace uso de los mismos algoritmos de predicción y su enfoque es la mejora de la calidad del servicio mientras se mantienen bajos costos de recursos. En este trabajo se propone adicionalmente una función de medida de costo-sensitiva llamada CPC (Cloud Prediction Cost) para guiar el proceso de predicción.

Si bien estos artículos se relacionan a la gestión de capacidad de infraestructura tecnológica haciendo uso de técnicas de Inteligencia Artificial, el enfoque de abordaje es diferente al del presente trabajo ya que su aplicación está orientada a la provisión de infraestructura en la nube, mientras que el trabajo que se desarrolla en este documento está dirigido a los consumidores de infraestructura, sin limitarse a la nube,

es decir, también contempla el uso de infraestructura local de equipos instalados en sus propios centros de datos (no necesariamente nube privada).

Al momento de implementar el modelo de inteligencia artificial, se tomará en consideración los algoritmos de predicción de regresión lineal, máquinas de vector de soporte y redes neuronales propuestos en estos artículos.

- En el artículo “Artificial Intelligence-based 5G network capacity planning and operation”, los autores Pérez-Romero, Sallent, Ferrús y Agustí (2015) exponen como un gran desafío para los operadores de redes de comunicaciones 5G, el satisfacer eficientemente los requerimientos futuros de este tipo de redes. Plantean la revisión de las metodologías que actualmente se usan para la planeación y operación de redes incorporando conceptos de Inteligencia Artificial como soporte para la toma de decisiones. El planteamiento de este caso difiere del presente trabajo, ya que está destinado a la planeación de capacidad para proveedores de comunicaciones en 5G, mientras que el trabajo de este documento se orienta al análisis de capacidad de infraestructura tecnológica de centro de cómputo, principalmente servidores, no comunicaciones móviles 5G.

La mayor cantidad de trabajos de aplicación de Inteligencia Artificial a la gestión de la demanda y/o gestión de capacidad está dado en otras áreas como por ejemplo la energía, entre los cuales se tienen los siguientes:

- En el artículo “Computational Intelligence Approaches for Energy Load Forecasting in Smart Energy Management Grids: State of the Art, Future Challenges, and Research Directions”, Seyedeh, Deo, Shojafar, Mohammad, Conti y Shamshirband (2018) explican sobre la importancia de la gestión de la demanda en el sector eléctrico y explora el estado del arte en cuanto a métodos de aprendizaje automático para realizar predicciones de carga para clasificar y evaluar operaciones sustentables en la gestión del sistema eléctrico. En este documento se analizan diferentes técnicas de Inteligencia Artificial yendo desde las redes neuronales de perceptrones multicapa y deep learning hasta las estrategias de colonias de abejas inteligentes y algoritmos de clusterización como K-shape. Finalmente, se presentan los criterios que se utilizan para evaluar predicciones inteligentes de carga y se hace un compendio de los resultados de evaluaciones realizadas previamente por otros autores sobre algunos de los algoritmos revisados. Se considerará las redes neuronales dentro de la selección de técnicas para la generación de pronósticos y la evaluación de las mismas.

- Huang, Tang y Chen (2018) en la publicación “Energy Demand Forecasting: Combining Cointegration Analysis and Artificial Intelligence Algorithm” presentan un caso de generación de pronósticos sobre la demanda de energía eléctrica en China utilizando algoritmos de aprendizaje automático e integrándolos con métodos econométricos que incluyen el análisis de cointegración, con el cual se establece una relación de largo plazo entre las variables y que a partir de pruebas de cointegración se puede validar la confiabilidad de las proyecciones. En la implementación se hace uso de redes neuronales artificiales, con las cuales se realizan las proyecciones. Para afinar el modelo, se determina si existe una relación entre la demanda de energía y los factores considerados, en caso de no existir esta relación se procede a ajustar el modelo. Al igual que el estudio anterior esta investigación servirá como referencia para la implementación de proyecciones de capacidad utilizando redes neuronales artificiales.
- En el libro “Advanced Solutions in Power Systems” de Eremia, Liu y Edris (2016, pp. 2-7) se afirma que el poder y versatilidad de las técnicas de Inteligencia Artificial (IA) se aplican a todos los niveles de la operación, control, proyecciones y actividades de planificación de los sistemas de energía, y permiten que estos evolucionen hacia redes inteligentes (smart grids). Entre las principales técnicas de IA aplicadas en este campo se mencionan los algoritmos genéticos, redes neuronales artificiales, sistemas expertos, lógica difusa, y los árboles de decisión. De manera particular, se menciona que se han utilizado redes neuronales artificiales para predecir la carga en sistemas de energía.
- Veit, Xu, Zheng, Chakraborty, Sycara (2014) en su trabajo “Demand Side Energy Management via Multiagent Coordination in Consumer Cooperatives” proponen el uso de coordinación multiagente para gestionar de forma más eficiente la demanda de energía eléctrica en consumidores. En este caso las personas, usuarias de la electricidad, reciben retroalimentación adicional de los precios de consumo en función de una demanda coordinada entre diferentes consumidores de manera que puedan modificar su patrón de demanda y minimizar el costo de la energía eléctrica. Este trabajo no contempla la generación de proyecciones de la demanda o de la capacidad de energía eléctrica.

En resumen, las referencias de Inteligencia Artificial aplicadas a la gestión de la capacidad, que han sido identificadas como estado del arte, tienen un enfoque diferente al planteado en el presente proyecto, como se puede observar en la *Tabla 1*.

Tabla 1. Estado del arte y relación con el presente trabajo

Estado del arte	Diferencia con enfoque del presente trabajo
(Jiang et al., 2012) (Jiang et al., 2013)	Estos trabajos están enfocados en gestión de la capacidad para la mejora de la calidad del servicio y la eficiencia de costos en proveedores de infraestructura en la nube. Por otro lado, el presente trabajo se orienta al análisis de capacidad de infraestructura para organizaciones que consumen infraestructura sea a través de la nube o de forma local. Al analizarse del lado del consumidor se cuenta con información adicional de la demanda que será utilizada en los modelos predictivos y que no fueron consideradas en los estudios indicados.
(Pérez-Romero et al., 2015)	Está relacionado con el sector de las telecomunicaciones y la planeación de capacidad para operadores de redes de comunicaciones 5G. El presente trabajo se orienta a organizaciones que utilizan recursos de infraestructura y redes no como proveedores de estos productos. Adicionalmente, se contempla otro tipo de infraestructura de centro de cómputo como son los servidores. No se contempla el análisis de infraestructura de comunicaciones móviles 5G.
(Seyedeh et al., 2018) (Huang et al., 2018) (Eremia et al., 2016)	Están orientados a la aplicación de la Inteligencia Artificial a la gestión de demanda en el sector eléctrico y no la gestión de capacidad de la infraestructura tecnológica como se plantea el presente trabajo.
(Veit et al., 2014)	En este estudio se aplica la Inteligencia Artificial a la gestión de la demanda de energía eléctrica, con un enfoque de coordinación multiagente el cual provee información a los consumidores del suministro eléctrico. El presente trabajo no contempla algoritmos de coordinación multiagente, en su lugar se utilizará la Inteligencia Artificial para la proyección de capacidad de recursos tecnológicos y se realizará de manera centralizada.

Fuente: Elaboración propia

A continuación, se describen los objetivos planteados en el presente trabajo y metodología utilizada.

3 Objetivos y metodología de trabajo

En este capítulo se presentan los objetivos planteados en torno a la aplicabilidad de la Inteligencia Artificial al campo de la Gestión de Capacidad de Infraestructura Tecnológica. Así también se detalla la metodología CRISP-DM seleccionada para la implementación de la solución.

3.1 Objetivo general

Facilitar la gestión de la capacidad de la infraestructura tecnológica a través de la provisión de una herramienta inteligente que permita pronosticar los requerimientos de capacidad con un error menor al 20% del valor máximo de la métrica.

3.2 Objetivos específicos

Para alcanzar el objetivo general antes planteado, es necesario también establecer objetivos intermedios mismos que se listan a continuación y que han servido como guía para el desarrollo del presente trabajo.

- Establecer un caso de aplicación para el análisis de la capacidad de infraestructura tecnológica.
- Implementar el monitoreo de la infraestructura y recolectar datos.
- Establecer los requerimientos del sistema en base al caso de aplicación.
- Analizar los datos recolectados y prepararlos para los modelos.
- Identificar las técnicas de aprendizaje supervisado más adecuadas e implementar los modelos de inteligencia artificial.
- Evaluar la solución planteada con base al caso de aplicación definido.
- Implementar la interfaz de uso del modelo.

A continuación, se describe la metodología utilizada para la implementación.

3.3 Metodología CRISP-DM

Para la implementación de la solución de inteligencia artificial aplicada a la gestión de capacidad de infraestructura tecnológica se utilizará la metodología CRISP-DM (CRoss-Industry Standard Process for Data Mining). Si bien esta metodología nació en el año 1996

bajo el concepto de la minería de datos, es perfectamente aplicable a la implementación de soluciones de aprendizaje automático ya que cuentan con una base común que es el tratamiento de información. La solución propuesta en el presente trabajo contempla como elemento principal un componente de predicción de requerimientos de infraestructura tecnológica; y de acuerdo a Kelleher, Namee, D'Arcy (2015), la metodología CRISP-DM es recomendada para proyectos de análisis predictivo de datos ya que incrementa la probabilidad de éxito. En el libro de los autores antes referidos, se utiliza la metodología CRISP-DM junto con técnicas de aprendizaje automático para el análisis predictivo, lo cual refuerza el hecho de que esta metodología es aplicable a la solución de inteligencia artificial planteada. Adicionalmente, los autores Wirth y Hipp (2000) mencionan que este modelo de proceso persigue que los proyectos del tipo minería de datos sean menos costosos, más confiables, más repetibles, más manejables y rápidos; por lo que estos beneficios se aplicarían también al presente trabajo.

De acuerdo a Chapman et al. (2000), la metodología CRISP-DM cuenta con cuatro niveles que van de mayor a menor nivel de detalle y que son: las fases, las tareas genéricas, las tareas especializadas y las instancias de proceso. Para la implementación de la solución de inteligencia artificial planteada se utilizarán principalmente las fases definidas en esta metodología, las tareas y demás detalles se ajustarán al caso particular del proyecto. En la Figura 3 se puede apreciar gráficamente cada una de las fases definidas en la metodología CRISP-DM.

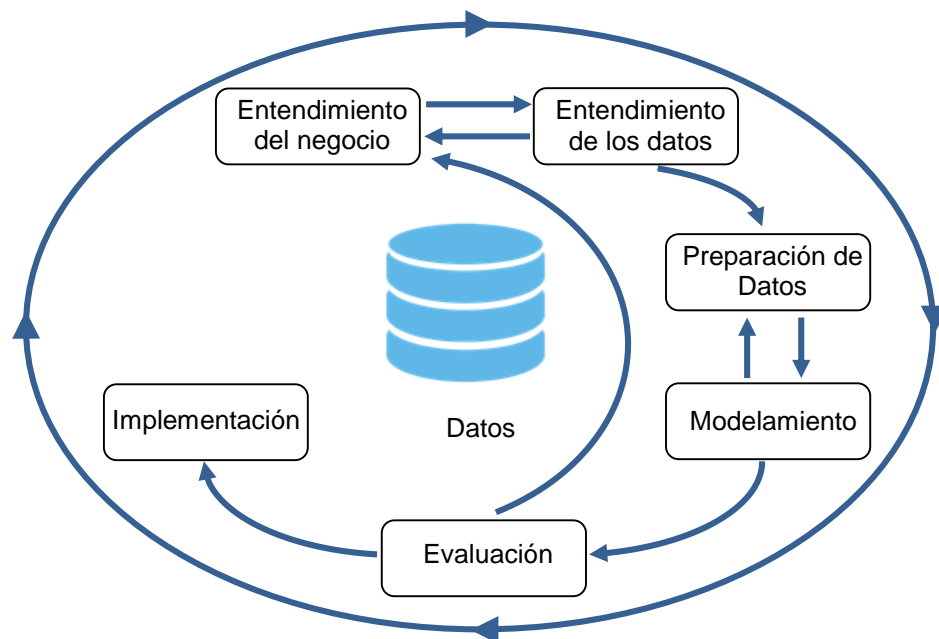


Figura 3. Fases de la metodología CRISP-DM.

Esta figura muestra las fases de la metodología seleccionada para la implementación de la solución
Fuente: Basado en (Chapman et al., 2000)

Chapman et al. (2000), describe 6 fases principales en la metodología CRISP-DM y las flechas presentadas entre fases en la Figura 3 indican las dependencias más frecuentes e importantes entre ellas; sin embargo, se explica que la secuencia no es rígida y puede existir interacción hacia adelante y atrás en cada fase. La metodología también expresa un ciclo continuo alrededor del gráfico en donde la solución implementada puede dar paso a nuevas necesidades de negocio.

A continuación, basado en Chapman et al. (2000), se describe cada una de las fases y se explica la adaptación de autoría propia que se realizó respecto a las tareas de cada fase para la implementación del presente trabajo:

1. **Entendimiento del negocio:** en esta etapa inicial se establece la definición del problema a partir de los objetivos y requerimientos identificados desde la perspectiva de negocio. Manasson (2019) menciona que uno de los errores más comunes en los proyectos es iniciar sin tener bien definido el problema de negocio, es por eso que en esta etapa se clarifican dichos objetivos y necesidades. Adicionalmente, en esta fase se diseña también un plan preliminar para alcanzar los objetivos.

Considerando que la gestión de capacidad de infraestructura tecnológica es muy amplia, es necesario delimitar la implementación definiendo un caso específico de aplicación. En esta fase se definen, desde una perspectiva de usuario final, los objetivos y requerimientos específicos para el caso de aplicación de la solución de inteligencia de negocios a implementar. Es así que, durante esta etapa, se establece el alcance de la gestión de capacidad tecnológica del caso y se detallan: las métricas, el entorno y componentes de infraestructura específicos en base a los cuales se deberá generar las predicciones de capacidad.

Durante esta etapa también se definen los requerimientos del sistema inteligente, para lo cual se utiliza un diagrama de casos de uso UML (Unified Modelling Language) el cual expresa las principales funcionalidades que estarán disponibles al usuario final. El diagrama se acompaña de una explicación textual de las necesidades funcionales de los casos de uso.

2. **Entendimiento de los datos:** durante esta fase se realiza una recolección preliminar de datos y se realiza un primer análisis e identificación de la información con la que se cuenta. Se identifican también posibles problemas de calidad de datos y se determinan los subconjuntos de información de mayor interés. De acuerdo a Manasson (2019), entre las principales tareas que contemplan esta fase están: determinar qué data es necesaria, recolectar los datos (si no están disponibles), explorar los datos y verificar

su calidad. Como resultado de esta fase, es posible conocer las características que se van a tener como entrada, así como sus tipos de datos, con lo cual se puede empezar la selección de características importantes y el formateo de la información.

A efectos del presente trabajo, se contempla para el caso de aplicación el análisis de los recolectores de datos y la identificación de la data con la cual se cuenta para la implementación del modelo.

- 3. Preparación de Datos:** en esta fase se realizan las implementaciones necesarias que permiten obtener el conjunto de datos final. Va desde la selección de atributos hasta la transformación y limpieza de datos, dejando lista la información que necesita el modelamiento. Manasson (2019) explica que entre las decisiones que se toman en esta etapa están el tratamiento de los datos faltantes (missing values), anomalías (outliers) y casos atípicos.

Para el proyecto, durante esta etapa, se prevé la selección final de atributos a utilizar y la implementación de mecanismos que automaticen la recolección de datos y la transformación de las entradas a un formato estándar con el que trabajará el modelo.

- 4. Modelamiento:** durante esta etapa se seleccionan las técnicas a utilizar y se afinan sus parametrizaciones. En función de las técnicas seleccionadas es frecuente que se requiera retornar a la preparación de datos.

En el proyecto, se contempla durante esta etapa la definición de la arquitectura de la solución y la selección e implementación de los modelos de Inteligencia Artificial para la proyección de requerimientos de capacidad de la infraestructura tecnológica de acuerdo al caso de aplicación. Se contempla el uso de tecnologías de aprendizaje supervisado y el afinamiento de los modelos en base a los resultados obtenidos durante el entrenamiento.

- 5. Evaluación:** se enfoca en evaluar tanto el modelo generado en la fase anterior como los pasos seguidos para su construcción, con el fin de garantizar que se cumplan los objetivos establecidos en la primera fase. Finalmente, en esta fase se decide el uso de los resultados alcanzados por el modelo. De acuerdo a Manasson (2019) para proyectos que generan modelos de predicción de datos, en esta etapa se evalúa el rendimiento del modelo a partir de los datos de prueba.

Para el proyecto, en base a la información histórica se contempla un conjunto de datos para pruebas a fin de validar la calidad de los modelos implementados. Finalmente, se

definirán y calcularán métricas de evaluación que permitirán determinar si los modelos desarrollados alcanzaron los objetivos planteados en el caso de aplicación.

6. **Implementación:** en esta fase, se debe organizar y generar la presentación del conocimiento provisto por el modelo de forma que este le sea de utilidad al cliente. Manasson (2019) afirma que la estrategia de despliegue debe asegurar la legibilidad, reproducibilidad y mantenibilidad del análisis implementado en el modelo, para lo cual se debe considerar el detallar y documentar tanto las decisiones tomadas respecto al modelo como cada paso de la implementación; adicionalmente, el autor considera necesario detallar en esta etapa las instrucciones de instalación y uso de la solución.

En esta fase final del proyecto, se validará que se encuentren debidamente documentados cada uno de los modelos. Adicionalmente, se implementarán gráficos explicativos que faciliten el análisis de los datos por parte del usuario final y se desarrollarán las guías de instalación y uso de la solución.

Con el fin de brindar mayor claridad para el presente trabajo a esta etapa se la denominará "Instalación e interfaz de uso".

4 Implementación de la solución de IA para la gestión de la capacidad tecnológica

A la solución de inteligencia artificial desarrollada se la nombró **IACT** por ser una herramienta de **Inteligencia Artificial** aplicada a gestión de la **Capacidad Tecnológica**. La implementación se realizó aplicando la metodología antes presentada. Cada subcapítulo corresponde a la descripción de la ejecución de cada fase de acuerdo a la metodología CRISP-DM.

4.1 Entendimiento del negocio – caso de aplicación

Debido a que la gestión de la capacidad es un proceso extenso y complejo y a que los tipos de componentes de infraestructura son muy variados, en este apartado se establece el caso de aplicación sobre el cual se realizará la implementación de la solución de proyección de requerimientos de capacidad de infraestructura tecnológica. En base a este caso de aplicación, se detallarán a continuación los requerimientos del sistema.

4.1.1 Definición del caso de aplicación – administrador de libros

La realización del presente trabajo involucra dos retos principales que se detallan a continuación:

- 1. Diversidad de la infraestructura:** En cuanto a la infraestructura tecnológica se tiene una infinidad de componentes de hardware y software que podrían ser analizados. Los dispositivos varían de una marca a otra y su arquitectura de uso difiere de organización a organización.
- 2. Necesidad de información:** Para el desarrollo de modelos de Inteligencia Artificial es indispensable contar con la información suficiente que permita identificar patrones de comportamiento y entrenar los modelos para que puedan realizar las predicciones. Esta información pertenece a cada organización y no es de acceso público.

A fin de superar estos dos retos y viabilizar la construcción de una herramienta de Inteligencia Artificial (IA) aplicada al campo de la Gestión de Capacidad de Infraestructura Tecnológica, se optó por construir un caso específico de aplicación, es decir, se implementó un sistema que procesa transacciones ficticias y que genera consumo de recursos de infraestructura. Esta estrategia permite:

- Reducir la diversidad de infraestructura y enfocarse a componentes concretos que forman parte del caso de aplicación implementado. El análisis de la capacidad se realizará únicamente sobre los componentes definidos para el caso.
- Recolectar la información necesaria de la demanda y del consumo de recursos en los servidores, lo cual servirá de insumo para la implementación de los modelos de aprendizaje automático.
- Validar la aplicabilidad de la solución sobre un caso concreto.

En la siguiente sección se describe las funcionalidades consideradas para el caso de aplicación.

Funcionalidad del caso de aplicación:

Es importante considerar que un servicio tecnológico solventa una necesidad de negocio y dicha necesidad es la fuente de información de la gestión de la demanda que alimenta a su vez a la gestión de la capacidad de la infraestructura tecnológica. Para el caso de aplicación se decide tomar, como temática de negocio, la administración de un inventario de libros. La administración se realizará a través de una aplicación web que contempla las transacciones de: registro, actualización, consulta y eliminación (lógica) de libros. A continuación, se describe cada funcionalidad expuesta por la aplicación web a mayor detalle.

- 1. Registro de un libro:** en esta transacción se reciben los datos de un libro: título, autor y resumen, y se procede a generar un código único y a almacenarlo en el sistema junto con el estado activo.
- 2. Actualización de datos de un libro:** en esta funcionalidad se recibe como entrada el código del libro que se desea actualizar y los nuevos datos de: título, autor y resumen. Con esta información se procede a registrar los cambios en el repositorio.
- 3. Eliminación de libros:** consiste en una eliminación lógica de los libros modificando su estado de activo a pasivo. Se recibe un código de libro y se actualiza su estado a inactivo.
- 4. Consulta de libros:** dado un texto que se desea consultar, se buscan todos los libros que tengan esa palabra dentro del título del libro.

Adicionalmente, para los cuatro tipos de transacciones se debe almacenar un registro con la fecha y hora en que se realizó cada transacción. Esta información será de vital importancia para poder analizar la relación que exista entre la demanda y la capacidad de recursos de infraestructura utilizados.

Una vez definida la funcionalidad, es necesario establecer el patrón de uso que tendrá el sistema. Estas definiciones se detallan en el siguiente apartado.

Definición de la demanda simulada para el caso de aplicación:

En el uso de los sistemas pueden existir ciertos patrones de comportamiento, al igual que sucede en la atención presencial de clientes. Así, por ejemplo, si se piensa en un supermercado, probablemente durante las primeras horas del día la afluencia de clientes sea menor que durante la tarde y noche ya que en este último período muchos trabajadores han concluido su jornada laboral y acuden a realizar compras. Si se piensa en períodos más extensos se podría identificar que existe mayor demanda en ciertos meses como por ejemplo en diciembre en donde debido a las festividades existe una mayor cantidad de compradores. Por otro lado, en sistemas expuestos a través del internet, aun cuando se brinde una disponibilidad de 7x24 (7 días a la semana 24 horas al día) se podrán identificar horarios de mayor y menor uso que obedece a patrones de uso de los clientes. Por ejemplo, un sistema bancario web podría tener baja transaccionalidad en horas de la madrugada por cuanto la mayoría de clientes se encuentran durmiendo.

En base a lo antes expuesto, para el caso de aplicación se decide simular la siguiente transaccionalidad mediante 5 procesos concurrentes en donde cada uno de ellos ejecuta un número de transacciones de cada tipo. Para la definición del número de transacciones que ejecuta cada proceso se considera tanto un patrón en base a una función como cierto grado de variación en el comportamiento utilizando números aleatorios:

- **Demanda para el registro de libros:** para el registro de libros se define de manera arbitraria el patrón de demanda que se aprecia en la *Figura 4*:

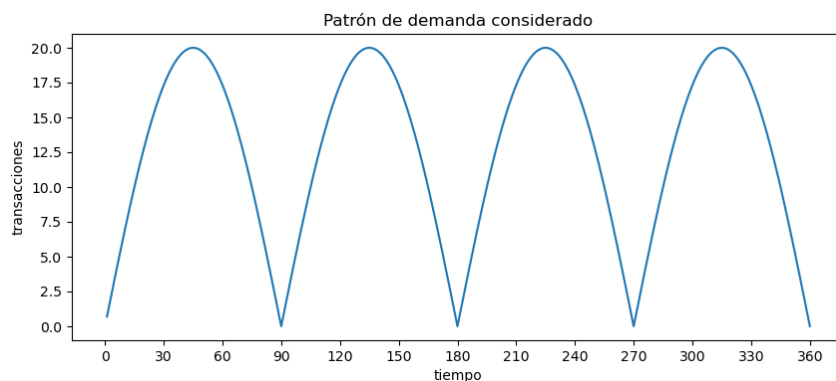


Figura 4. Patrón de demanda para el caso de aplicación – registro de libros.

En esta figura se puede apreciar la función considerada como patrón de demanda para el registro de libros

Fuente: Elaboración propia

Para generar este patrón se utilizó una función sinusoidal con la siguiente fórmula:

$$Y = \sin \frac{\pi * (X \% P)}{P} * M$$

En donde:

- Y: es el número de transacciones a ejecutar
- X: es un número secuencial que representa un instante de tiempo
- P: es el período, es decir el número de instantes de tiempo cada cuanto se repetirá el patrón. En el gráfico se aprecia un P=90.
- M: es el número máximo de transacciones a ejecutar. En el gráfico se aprecia un M=20

A este patrón se lo complementa añadiendo un valor aleatorio de transacciones entre 0 y 3 y se obtiene el patrón de la *Figura 5*.

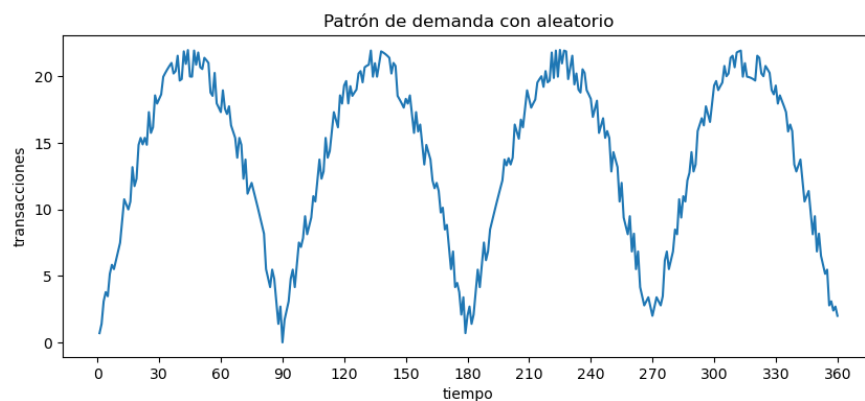


Figura 5. *Patrón de demanda con factor aleatorio para registro de libros.*

En esta figura se puede observar el patrón de demanda considerando un factor aleatorio.
Fuente: Elaboración propia

- **Demanda para las otras transacciones:** para los tres tipos de transacciones restantes se consideró una demanda en donde el número de transacciones fue completamente aleatorio:
 - Actualización: aleatorio entre 0 y 8 transacciones.
 - Eliminación: aleatorio entre 0 y 5 transacciones.
 - Consulta: aleatorio entre 0 y 10 transacciones.

Otro elemento importante es contar con información para simular las transacciones. Esto se explica a continuación.

Información a utilizar para demanda simulada:

Para simular las transacciones se requería gran cantidad de información y con el fin de que el proceso sea lo más cercano posible a un caso real se buscaron fuentes de información reales. La alternativa que se consideró más idónea por el volumen y facilidad de acceso a la información fue el Proyecto Gutenberg, el cuales una biblioteca de libros gratuitos que incluyen obras antiguas de los Estados Unidos, cuyos derechos de autor (copyright) han expirado (Project Gutenberg, 2020). Accediendo al sitio web y navegando entre sus opciones, se identificó el siguiente enlace: http://www.gutenberg.org/wiki/Gutenberg:Offline_Catalogs, de donde se obtuvo un listado en formato de archivo de texto de más de 60.000 nombres de libros con sus respectivos autores. A partir de esta información se implementó la simulación de transacciones de registro y actualización de libros, como se explica a continuación.

4.1.2 Implementación del administrador de libros

La implementación de la solución de administración de libros, que es el caso de aplicación de la gestión de capacidad, parte de los requerimientos presentados en los apartados anteriores y para su desarrollo se siguieron los siguientes pasos:

- Preparación de datos, consiste en el análisis del contenido de los datos obtenidos del Proyecto Gutenberg y las acciones de transformación que fueron necesarias realizar para que la información pueda ser utilizada por el simulador de carga transaccional.
- Definición de la arquitectura, en la cual se describe los componentes y tecnologías utilizadas para el caso de aplicación.
- Implementación, se desarrolló el aplicativo web en base a la arquitectura establecida. Adicionalmente se implementó un aplicativo java cliente, el cual permitió simular la carga transaccional sobre la aplicación web.
- Ejecución y toma de métricas, durante la ejecución del aplicativo web y cliente se realizó la toma de las métricas de capacidad las cuales permiten entrenar el modelo inteligente.

A continuación, se explica a mayor detalle cada etapa.

Preparación de datos:

En primer lugar, se realizó una revisión de la información obtenida del catálogo de Gutenberg y se identificó que se tenían varios bloques por año con la estructura de la *Figura 6*:

```

=====
                                GUTINDEX.2014

GUTINDEX.2014 is a plain text listing of eBooks posted to the Project
Gutenberg collection between January 1, 2014 and December 31, 2014
beginning with eBook number 44557 and ending with eBook number 47836.

**** A C Following a Project Gutenberg eBook Number Indicates Copyright ****

**** The Language of the eBooks is English, unless otherwise noted ****

~ ~ ~ Posting Dates for the below eBooks:  1 Dec 2014 to 31 Dec 2014 ~ ~ ~

TITLE and AUTHOR                                                    EBOOK NO.

The Matron's Manual of Midwifery, and the Diseases of Women          47836
During Pregnancy and in Childbed, by Frederick Hollick
[Subtitle: Being a Familiar and Practical Treatise,
 more especially intended for the Instruction of
 Females themselves, but adapted also for Popular
 Use among Students and Practitioners of Medicine]

Gróf Kaczaifalvi László avagy a természetes ember, by Ferencz Versegly 47835
[Illustrator: Elemér Kőszeghy-Winkler]
[Language: Hungarian]

```

Figura 6. *Ejemplo de información de libros utilizada para el caso de aplicación.*
 Esta captura permite entender el trabajo de preparación de datos que se debió realizar.
 Fuente: *archivo obtenido de http://www.gutenberg.org/wiki/Gutenberg:Offline_Catalogs*

Como se puede observar por cada año se tiene un conjunto de cabeceras y los datos del libro se encuentran en varias filas. Adicionalmente, se puede apreciar que al final de la primera fila de cada libro se tienen varios espacios y un número que corresponde a un secuencial del número de libro.

Para poder extraer un listado de libros más limpio se eliminaron manualmente las primeras y últimas líneas del catálogo dejando únicamente el contenido que lista los archivos. Fue necesario además crear un programa Java denominado `PreprocesaArchivo.java`, el cual estructuró el contenido en una sola línea por tema de libro. Este programa entre otras cosas realiza las siguientes tareas:

- Eliminar las cabeceras, se elimina todo texto que se encuentre entre las líneas con caracteres "===" y "~ ~ ~". Se eliminan también las filas que empiezan con "TITLE".
- Se eliminan líneas en blanco ya que se crea una sola línea por cada libro.
- Se eliminan los espacios y el número de libro que se encuentra al final de la primera fila, en el ejemplo antes presentado se eliminan los siguientes números: "47836" y "47835"
- Se concatenan las filas que tienen un espacio al inicio con el texto del libro anterior.

Finalmente, se obtuvo un nuevo archivo denominado listado_libros.txt el cual es utilizado por el cliente del aplicativo web para simular el registro y actualización de libros. El código del programa se encuentra publicado conforme a lo explicado en el Anexo I. Código .

Definición de la arquitectura del caso de aplicación:

Para la automatización de la administración de libros (caso de aplicación) se estableció una arquitectura distribuida en tres capas: cliente, aplicación web y base de datos. Esta arquitectura simula un escenario típico de aplicaciones web empresariales y permite realizar la medición de consumo de recursos de infraestructura sobre los servidores tanto de aplicación como de base de datos.

A continuación, en la Figura 7 se presenta la arquitectura del caso de aplicación desarrollado:

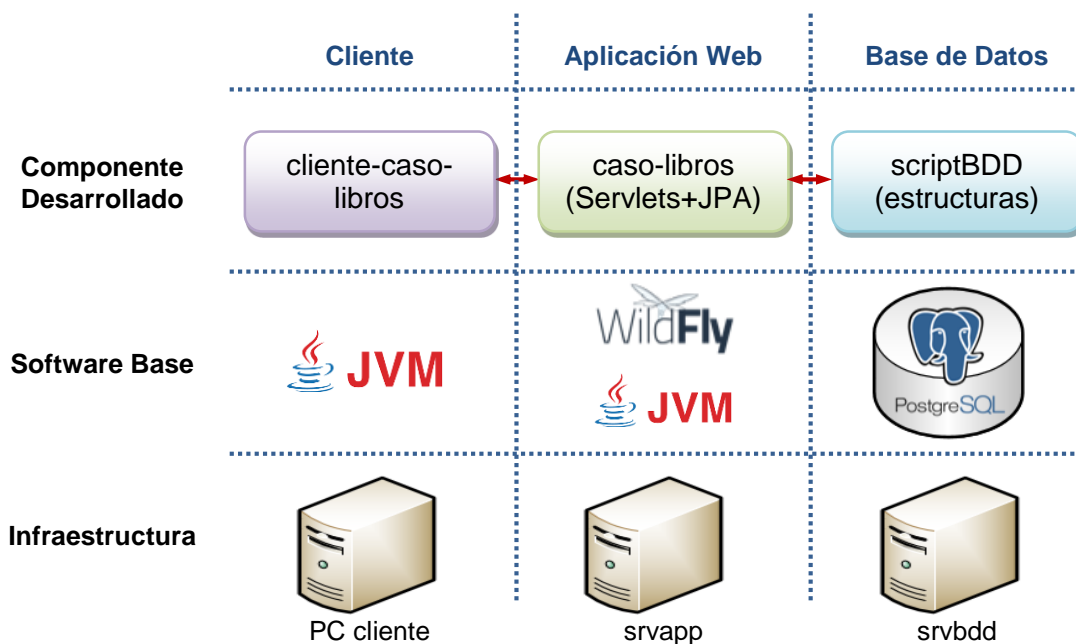


Figura 7. Arquitectura del caso de aplicación – administrador de libros

Esta figura muestra la arquitectura del caso sobre el cual se realizarán las predicciones de la capacidad de infraestructura utilizando Inteligencia Artificial.

Fuente: Elaboración propia

A continuación, se describe cada elemento de la arquitectura de izquierda a derecha y de arriba hacia abajo:

- 1. cliente-caso-libros:** corresponde al programa desarrollado para simular las acciones de usuarios, generando transacciones concurrentes sobre la aplicación web. En este componente se encuentra desarrollado en tecnología java e incluye la siguiente lógica:

- Clase `PreprocesaArchivoLibros`: realiza la limpieza y preparación de datos para el registro y actualización de libros a partir del archivo del Proyecto Gutenberg.
 - Clase `ClienteSimuladorTransacciones`: es el programa principal el cual se encarga de crear 5 hilos de ejecución independientes y de calcular el número de transacciones de cada tipo que deben ser ejecutadas por los hilos.
 - Clase `EjecutorHilo`: haciendo uso de la clase "Util", esta clase ejecuta las transacciones de registro, actualización, consulta y eliminación en un nuevo hilo de ejecución. Se utiliza el número de transacciones definidas por la clase del punto anterior y se alterna la ejecución entre los diferentes tipos de transacciones, es decir ejecuta un registro, después una actualización, una eliminación y una consulta hasta completar el total de transacciones requerido.
 - Clase `Util`: contiene la lógica para realizar las conexiones http al servidor web y para invocar un componente web enviando la información mediante el método HTTP POST.
2. **cliente-caso**: es la aplicación web y fue desarrollada utilizando tecnologías Java Enterprise Edition (JEE). Expone las diferentes funcionalidades de administración de libros a través de Servlets que reciben los datos enviados por el cliente mediante invocaciones HTTP. Esta aplicación se conecta a la base de datos mediante Java Persistence Api (JPA) y realiza las acciones que correspondan por cada tipo de transacción.
 3. **scriptBDD**: contiene las sentencias de borrado y creación de las tablas y secuencias necesarias para la aplicación. En la implementación se consideraron las siguientes estructuras:
 - *Tabla libro*: la cual almacena la información de un libro: id, titulo, autor, resumen, fecha_hora_creacion y estado.
 - *Tabla transaccion*: almacena el histórico de transacciones ejecutadas, incluye los siguientes campos: id, tipo, id_libro y fecha (con hora).
 - Secuencias `libro_id_seq` y `transaccion_id_seq` son las secuencias utilizadas para generar los id (códigos únicos) de las estructuras antes mencionadas.
 4. **JVM**: son las siglas de Java Virtual Machine quien es la encargada de ejecutar los aplicativos. Este componente se utiliza tanto para el cliente como para el servidor de aplicaciones.

5. **WildFly:** es un servidor de aplicaciones que brinda un entorno transaccional en el cual se pueden ejecutar las aplicaciones desarrolladas con tecnología Java Enterprise Edition (JEE). La versión utilizada es la 18.0.0. Final. Entre sus funcionalidades cuenta con un servidor web, sobre el cual funciona la aplicación desarrollada.

6. **PostgreSQL:** corresponde a una base de datos relacional para el almacenamiento de la información generada. Entre las bondades de este tipo de componentes está el contar con transacciones de tipo ACID, por las siglas en inglés de Atomicity, Consistency, Isolation, y Durability lo que quiere decir que se asegura que una transacción se realice o reverse de forma completa, mantenga la consistencia de los datos sin afectar otras transacciones, y que asegure la persistencia de la información. En este caso se ha optado por la base de datos de software libre PostgreSQL versión 10.13 y este repositorio contiene la información generada por la transaccionalidad del caso de aplicación.

En este servidor se crean las estructuras de datos que almacenan la información de las transacciones ejecutadas.

7. **PC cliente, srvapp, srvbdd:** son los equipos físicos utilizados para la ejecución de las transacciones y sobre los cuales se realiza la medición de los recursos de infraestructura. En la *Tabla 2* se resumen las características de los tres computadores utilizados:

Tabla 2. Características de computadores utilizados

PC cliente	Srvapp	Srvbdd
Procesador: Core i3 de 2.2 GHz	Procesador: AMD E2-7110 de 1.8 GHz	Procesador: Core i5 de 2.27 GHz
Memoria Ram: 12 GB	Memoria Ram: 4 GB	Memoria Ram: 4 GB
Sistema operativo: Windows 8.1 Pro	Sistema operativo: Windows 7 Pro SP1	Sistema operativo: Windows 7 Pro
		Disco de datos: 177 GB

Fuente: Elaboración propia

Capacidad de infraestructura a medir para el caso de aplicación:

Es necesario contar con un mecanismo que permita capturar métricas de uso de la infraestructura tecnológica. Para al análisis de capacidad de la Administración de Libros se debe considerar las siguientes métricas:

- Servidor de Aplicaciones (srvapp): se debe medir la cantidad de memoria y el porcentaje de procesamiento utilizados durante la carga transaccional. No se considera la medición del almacenamiento para este componente ya que no generará un consumo relevante. El consumo de almacenamiento se da principalmente en la base de datos ya que ahí se deposita la información.
- Servidor de Base de Datos (srvbdd): de manera similar al componente anterior es importante medir la cantidad de memoria y el porcentaje de procesamiento utilizados. A diferencia del servidor de aplicaciones, en este caso también se medirá la cantidad de almacenamiento utilizado.

Con el caso de aplicación claramente definido, se establecen a continuación los requerimientos para la solución IA aplicada a la gestión de capacidad.

4.1.3 Definición de requerimientos del sistema inteligente

A partir de los objetivos del presente trabajo y del alcance del caso de aplicación descrito en los puntos anteriores se definen los requerimientos para la herramienta de Inteligencia Artificial aplicada a la gestión de la capacidad de la infraestructura. Los requerimientos se encuentran estructurados en ocho casos de uso principales, los cuales se presentan en la *Figura 8*.

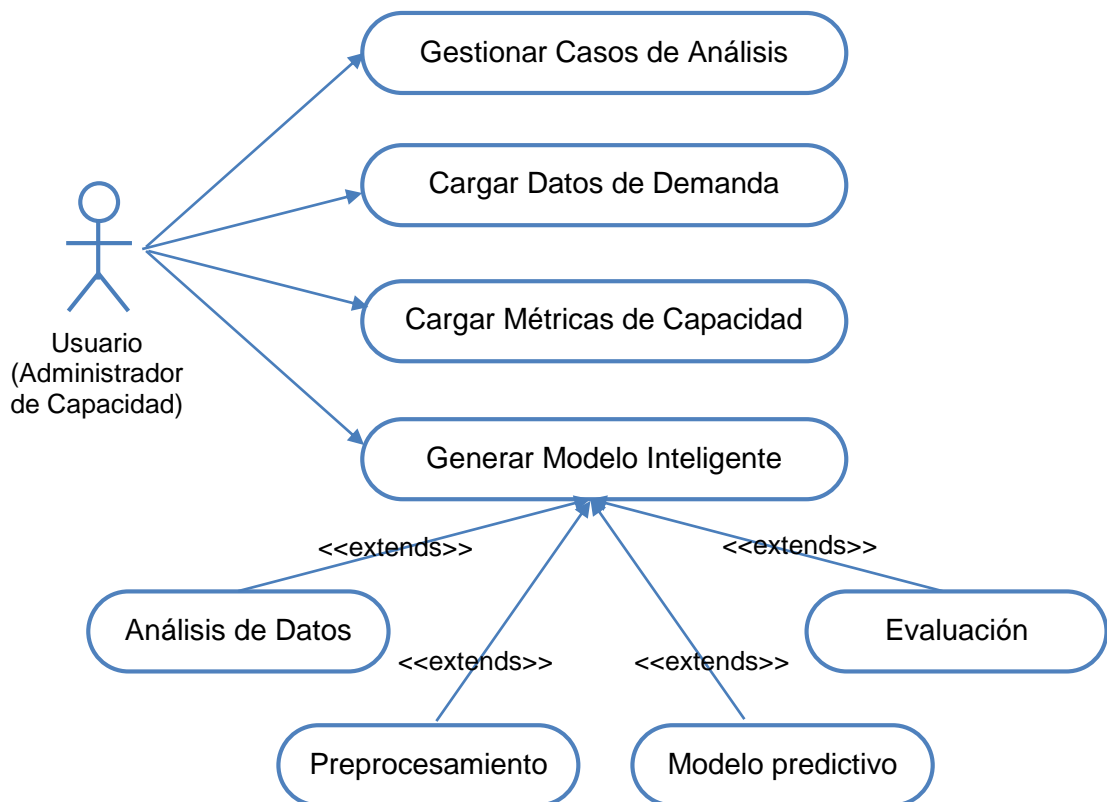


Figura 8. Casos de uso del sistema inteligente de gestión de capacidad
 En esta figura se presentan en formato UML (Unified Modelling Language) las principales funcionalidades a las cuales podrá acceder el usuario en el sistema inteligente.

Fuente: Elaboración propia

A continuación, se describe cada uno de los casos de uso:

- **Gestionar casos de análisis:** se considera como un caso de análisis a un conjunto de datos de demanda y métricas de capacidad relacionadas, para los cuales se realizará la generación del modelo de inteligencia artificial. Un claro ejemplo es el caso de aplicación de Administración de Libros, el cual contará con un conjunto de datos específicos tanto de demanda como de métricas de uso de infraestructura, para los cuales se generará el modelo inteligente.

Es necesario entonces considerar que el sistema inteligente permita crear más casos de análisis y gestionar cada uno de ellos de forma independiente. Así el usuario podría analizar datos de un tipo de infraestructura en un caso de uso y analizar otros recursos tecnológicos en otro caso. Esta funcionalidad también podría servir para analizar escenarios, en donde cada escenario sería un caso de análisis.

Dentro de la gestión de casos se debe contar con las siguientes funcionalidades:

- Creación de caso de análisis, permite que el usuario defina un nuevo caso.
 - Listado de casos de análisis, se presentan todos los casos de análisis existentes.
 - Selección de un caso de análisis, permite escoger el caso de análisis sobre el cual se va a trabajar. Una vez seleccionado un caso, sobre este se podrá cargar los datos y generar el modelo. Para acceder al resto de casos de uso el usuario necesariamente deberá haber seleccionado un caso.
 - Eliminar caso de análisis, permite eliminar casos que ya no sean requeridos.
- **Cargar Datos de Demanda**, la información de la demanda o carga transaccional es un insumo indispensable para la creación del modelo de inteligencia artificial. La carga se realiza para un caso de análisis específico. Con el fin de contar con esta información, se prevé dos funcionalidades principales:
 - Carga de archivos, se considera como formato estándar de datos para los modelos el uso de archivos CSV (valores separados por comas). Esta funcionalidad permitirá la carga de datos de demanda que se encuentren en archivos con este formato.
 - Obtención de datos desde la base de datos, como se comentó en el caso de aplicación Administración de Libros, la información será almacenada en tablas de la base de datos. Por lo tanto, para obtener la información de la carga transaccional de primera mano, es necesario contar con la funcionalidad que permita conectarse a la base, efectuar las consultas y finalmente generar el archivo CSV con la información.
- **Cargar Métricas de Capacidad**, este caso de uso permitirá alimentar al sistema con la información de las métricas de uso de recursos de infraestructura. Al igual que con la demanda, la carga de información se realiza para un caso de análisis específico. Se consideran las siguientes funcionalidades principales:
 - Carga de archivos CSV, permite subir al sistema aquellos datos que han sido capturados en formato de valores separados por comas.
 - Conversión de datos en otros formatos, en este caso el sistema permitirá pasar a archivos CSV aquellas métricas que se encuentran en un formato diferente. Se considerará la conversión para los formatos que se utilicen en la medición del caso de aplicación de Administración de Libros.

- **Generar Modelo Inteligente**, en este caso de uso se vincula la información de la demanda transaccional con las métricas de uso de recursos. Para lo cual permite:
 - Seleccionar los archivos del caso de aplicación que se desean analizar.
 - Por cada archivo seleccionar los campos (columnas) que serán considerados como índices y los demás campos que se incluirán en el modelo.
 - El usuario podrá escoger los tipos de algoritmos de aprendizaje automático que se incluirán en la generación.
 - Finalmente, se generará un modelo inteligente en base a la información de los puntos anteriores.

El modelo inteligente podrá ser revisado y ajustado por el usuario, en el caso de que lo considere necesario. Dentro del modelo generado se encontrarán los siguientes contenidos, que servirán de apoyo al usuario (administrador de capacidad):

- **Análisis de Información:** en esta sección se presentarán las características principales de los datos tanto de demanda como de métricas de capacidad.
- **Preprocesamiento de Datos:** contendrá todos los procesos necesarios que permitan cotejar la información de demanda y de capacidad, así como el tratamiento de datos faltantes (missing values).
- **Modelo predictivo:** corresponde a uno o varios modelos predictivos generados en base a las preferencias del usuario. Los modelos son implementaciones de algoritmos de aprendizaje automático.
- **Evaluación:** contempla un análisis de la calidad de los resultados obtenidos por el modelo, en base a datos de prueba.

En el siguiente apartado se presenta el análisis detallado de los datos de demanda y de métricas de capacidad de infraestructura que serán utilizados como entrada del modelo inteligente.

4.2 Entendimiento de los datos

Para la implementación de la solución se identifican dos tipos principales de datos:

- Demanda transaccional
- Métricas de uso de recursos de infraestructura

Cada uno se explica a mayor detalle a continuación.

Demanda transaccional

Como se explicó en la sección “2.1 La gestión de la capacidad tecnológica”, al momento de analizar la capacidad de la infraestructura tecnológica es necesario contar con la información de la demanda de negocio y entender su comportamiento.

Para el caso de aplicación la demanda corresponde a la cantidad de operaciones de registro, actualización, consulta y modificación de información de libros que se realizan. Esta información representa la necesidad actual de transacciones de negocio que debe soportar el sistema.

La fuente de esta información es la base de datos y en particular para el caso de aplicación se tomará la información que se encuentra en la tabla transacción y que detalla la fecha y hora de cada una de las transacciones.

Métricas de uso de recursos de infraestructura

De acuerdo a los requerimientos definidos, el sistema inteligente debe contar con un mecanismo que permita registrar las métricas de uso de la infraestructura tecnológica. Para cumplir con este requerimiento, se investigó diferentes alternativas y se definió la arquitectura de recolección de métricas presentada en la *Figura 9*

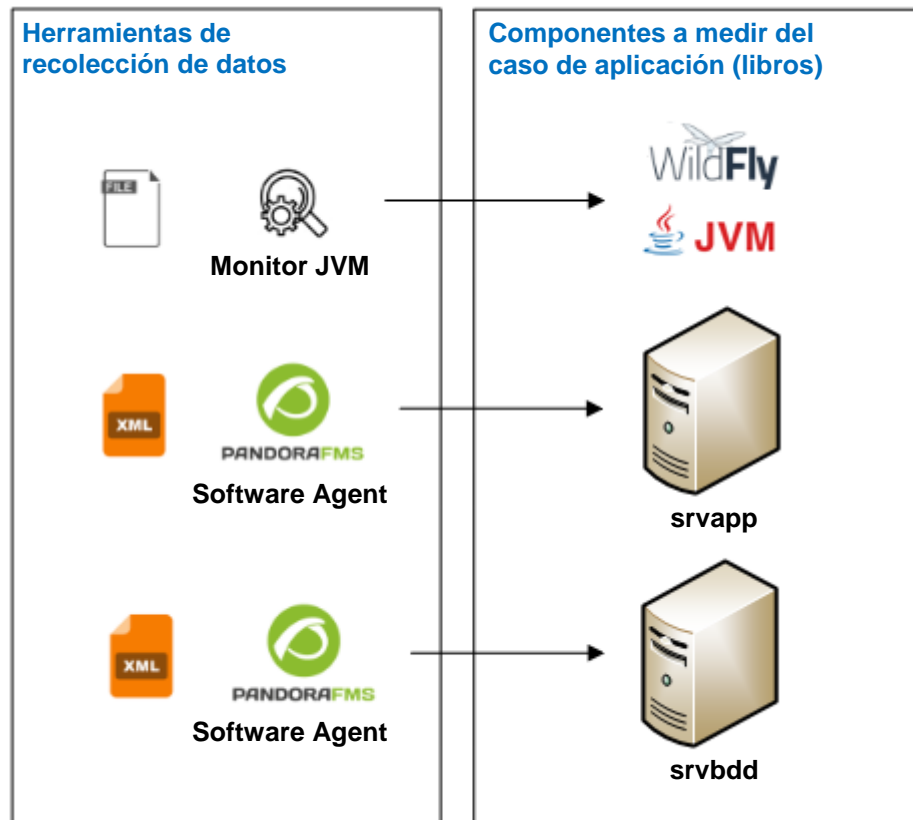


Figura 9. *Arquitectura de componentes de recolección de métricas*

Esta figura muestra la arquitectura de los componentes encargados de la recolección de métricas de consumo de recursos tecnológicos para el caso de aplicación (administrador de libros).

Fuente: Elaboración propia

A continuación, se describen los dos componentes de recolección de métricas:

- Monitor de Java Virtual Machine (JVM):** Las aplicaciones desarrolladas con tecnología Java corren sobre un entorno denominado Máquina Virtual de Java (JVM por sus siglas en inglés) la cual, entre otras cosas, interpreta el código compilado de la aplicación (archivos .class) y se encarga de gestionar la memoria de la aplicación. Cuando se inicia la ejecución de una JVM se pasan por parámetros el tamaño mínimo (-Xms) y máximo de memoria (-Xmx) y esta es administrada internamente, de tal manera que el programa nunca excederá el tamaño máximo. En el caso de que se haya utilizado toda la memoria asignada y que la misma haya quedado insuficiente se arroja un error denominado “java.lang.OutOfMemoryError”. Por esta razón resulta indispensable, para la gestión de la capacidad de infraestructura, el obtener el tamaño de memoria utilizado dentro de la JVM y no solamente a nivel del servidor.

El monitor implementado consiste en un componente web dentro de la aplicación, que se inicia al subir el sistema y registra de forma periódica el tamaño de memoria utilizado en la JVM para un tiempo dado. Este componente crea un nuevo archivo cada vez que sube el sistema. El nombre del archivo tiene el siguiente formato: yyyyMM-hhmiss-monitoreo-memoria.txt y su contenido se registra en formato CSV (valores separados por coma) con los siguientes datos: Fecha y hora, Memoria usada por la JVM(bytes).

- **Agente Software Pandora FMS:** Pandora FMS es un software de monitorización que puede ser utilizado en diferentes entornos y dispone de agentes que pueden ser instalados en sistemas operativos tanto de Microsoft como tipo Unix y Linux (Pandora FMS Enterprise, 2015). Pandora FMS cuenta con dos distribuciones una de código abierto y otra como versión empresarial. Entre sus componentes de software principales se tiene el servidor que permite la administración centralizada del monitoreo y los agentes que permiten recopilar métricas de los componentes que se monitorean. Para el presente trabajo, se utiliza el componte de código abierto denominado Agente Software de Pandora FMS, el cual se instala en los servidores que se desea monitorear y en base a la configuración de módulos de monitoreo genera archivos de tipo XML por cada captura de información incluyendo el detalle de los valores de métricas obtenidos.

El Agente Software Pandora FMS cuenta con un archivo de configuración denominado `pandora_agent.conf`, en el cual, entre los temas más relevantes, se puede definir el nombre del servidor que monitorea, la periodicidad con la cual se ejecuta y los módulos de las métricas que se van a registrar. Para el presente proyecto se configuró el agente para obtener las métricas de porcentaje de uso de CPU, porcentaje de memoria utilizada, total de memoria libre en MB (Mega Bytes). Adicionalmente, para el servidor en el que está instalada de la base de datos se mide el espacio disponible en disco en MB para analizar cómo influyen las transacciones en el almacenamiento.

El Agente Software Pandora FMS se ejecuta como un servicio y en base a la configuración definida, genera periódicamente archivos XML con la estructura presentada en la *Figura 10*.


```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<agent_data agent_name="srvapp" agent_alias="srvapp" description=""
  version="7.ONG.746(Build 200624)" timestamp="2020-08-03 05:22:21"
  address="192.168.0.4 , fe80::e11a:624a:5214:2f8e" interval="10"
  os_name="Microsoft Windows 7 Professional "
  os_version="Microsoft Windows 7 Professional Service Pack 1"
  group="Servers" parent_agent_name="" secondary_groups="" agent_mode="1">
<module>
  <name><![CDATA[CPU_Load]]></name>
  <type><![CDATA[generic_data]]></type>
  <description><![CDATA[User CPU Usage (%)]></description>
  <module_interval><![CDATA[1]]></module_interval>
  <min><![CDATA[0]]></min>
  <max><![CDATA[100]]></max>
  <min_critical><![CDATA[91]]></min_critical>
  <max_critical><![CDATA[100]]></max_critical>
  <min_warning><![CDATA[70]]></min_warning>
  <max_warning><![CDATA[90]]></max_warning>
  <unit><![CDATA[%]]></unit>
  <module_group>System</module_group>
  <data><![CDATA[4]]></data>
</module>
<module>
  <name><![CDATA[Memory_Used]]></name>
  <description><![CDATA[Used memory %]]></description>
  <data><![CDATA[48,92]]></data>
  <unit>%</unit>
  <min_critical>95</min_critical>
  <max_critical>100</max_critical>
  <module_group>System</module_group>
</module>
<module>
  <name><![CDATA[Free_Memory]]></name>
  <type><![CDATA[generic_data]]></type>
  <description><![CDATA[Non-used memory on system]]></description>
  <module_interval><![CDATA[1]]></module_interval>
  <data><![CDATA[1818]]></data>
</module>
</agent_data>

```

Figura 10. Ejemplo de información generada por el Agente Software Pandora FMS.

Esta captura presenta la estructura en la cual se genera la información de la herramienta de monitoreo. Con base a esto se establece el trabajo de procesamiento de datos para el modelo de inteligencia artificial.

Fuente: captura de uno de los archivos generados en el servidor srvapp

Como se puede observar, en la cabecera se obtiene la información general del servidor como por ejemplo el nombre y la IP, así también en el atributo timestamp se recibe la fecha y hora en que se realizó el monitoreo. A continuación, se presentan cada una de las métricas obtenidas dentro de las etiquetas denominadas “<module>”. Dentro de estas etiquetas entre los contenidos más relevantes se tiene el nombre de la métrica en la etiqueta “<name>”, la descripción, las unidades en que se mide (ejemplo: %) y en la etiqueta “<data>” se encuentra el valor de la métrica.

Relación entre los datos

Para poder realizar las estimaciones o predicciones de capacidad es necesario establecer un mecanismo para relacionar la información de las fuentes de datos antes indicadas. Se debe considerar que la información proviene de computadores y medios distintos. Se analizó y en base a los datos, se identificó que el mecanismo más adecuado es utilizar como enlace la fecha y la hora de los registros. De esta manera se puede establecer las métricas de los recursos que se utilizaron al mismo momento que se realizaron las transacciones de negocio.

Es importante mencionar que durante los primeros ejercicios de medición y de análisis de datos se identificó dos problemas que tuvieron que ser resueltos:

- **Sincronización de la hora de los computadores:** si los componentes no se encuentran sincronizados pueden existir inconsistencias en los datos. Así por ejemplo, si en el computador de la aplicación web se tiene un adelanto de 20 segundos respecto a la hora del computador de base de datos (tiempo t), al realizar la correlación por fecha y hora, se tendrían que para un número de transacciones concurrentes realizadas en el tiempo t las métricas que erróneamente se asumirían son las de transacciones que se realizaron 20 segundos antes, como se puede apreciar en la *Tabla 3*.

Tabla 3. Ejemplo de error por hora de computadores no sincronizada

Hora servidor BDD (t)	Transacciones realizadas	Hora servidor web (t+20)	Métrica consumo CPU (%)
8:00	20	8:20	4%
8:20	100	8:40	8%

Fuente: Elaboración propia

Para solventar este inconveniente se sincronizó a nivel de segundos las fechas y horas de los computadores.

En componentes de un ambiente real de Centro de Cómputo se recomienda la configuración de actualización de fecha en base a un servidor NTP (Network Time Protocol) el cual permite mantener la hora sincronizada.

- **Sincronización del formato de la hora en herramientas de monitoreo:** un inconveniente que se identificó al analizar los datos fue que una de las fuentes de información (Monitor JVM) almacenaba la fecha y hora en formato de 12 horas

mientras que el resto de monitores lo hacían en formato de 24 horas. Así en el un caso se tenía la hora 8:00 y en el otro 20:00.

Debido a que el componente Monitor JVM fue desarrollado, se realizó el cambio para que almacene la hora en el mismo formato del resto de métricas.

Se recomienda revisar siempre que los formatos de las fechas y horas de las herramientas de monitoreo sean iguales, caso contrario se deberán realizar las tareas de transformación necesarias.

Con el conocimiento de los datos se procede a la preparación de los mismos de forma que se facilite la ejecución de los modelos de inteligencia artificial. Estas tareas se describen en el siguiente punto.

4.3 Preparación de datos

La preparación de datos fue concebida en el sistema IACT, Sistema de Inteligencia Artificial desarrollado para la Gestión de la Capacidad Tecnológica, en dos etapas:

- Carga de datos y estandarización, el sistema permite cargar la información y en los casos que corresponde brinda la funcionalidad de transformación a un único formato de datos.
- Otras tareas de preparación, una vez homogenizados los formatos, se implementan en python las demás tareas de preparación.

En la *Figura 11* se presenta la arquitectura definida para la automatización de las tareas de preparación de datos:

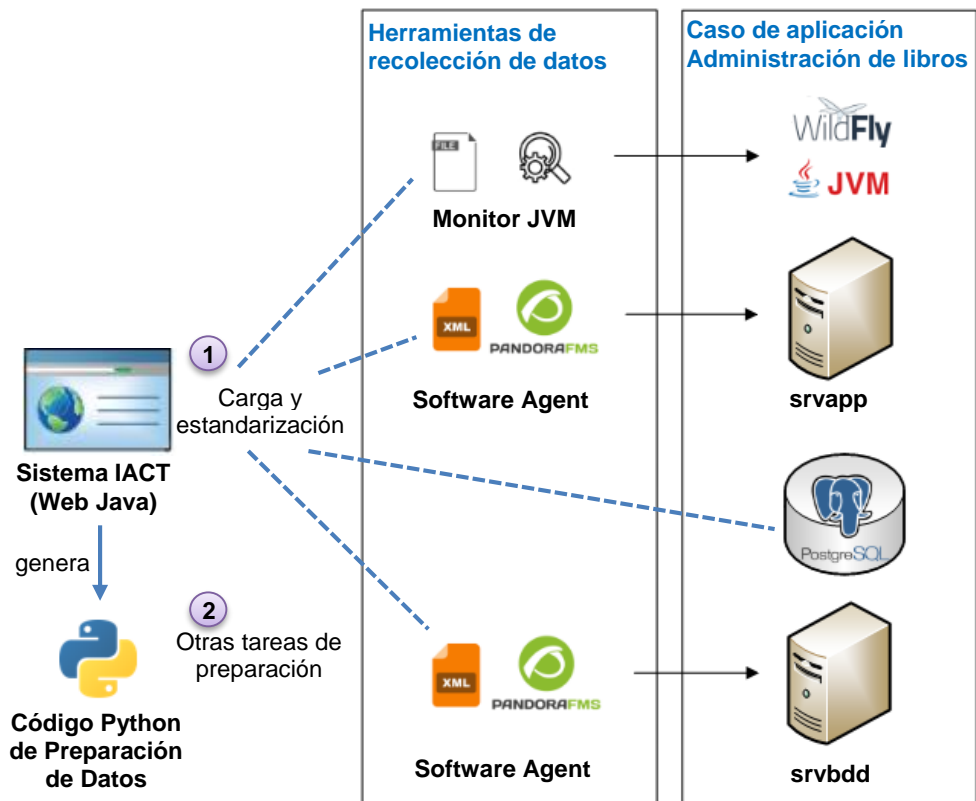


Figura 11. *Arquitectura de preparación de datos*

En la imagen se puede apreciar las dos etapas de preparación de datos consideradas por el sistema web Java IACT desarrollado.

Fuente: Elaboración propia

Como se puede apreciar, en la primera etapa se realiza la carga de las fuentes de datos al sistema IACT y este brinda las funcionalidades para transformar los datos a un formato estándar. En la segunda se genera el código python necesario para completar la preparación de los datos.

A continuación, se describe a mayor detalle las etapas de preparación de datos implementadas.

1) Carga y estandarización de datos:

Conforme a la arquitectura, esta etapa se automatiza a través de un aplicativo web desarrollado en lenguaje Java.

Como se pudo ver en el subcapítulo 4.2 *Entendimiento de los datos*, los datos para el análisis de capacidad del caso de Administración de Libros provienen de diferentes fuentes, entre las cuales se encuentran:

- Demanda transaccional, que se encuentra en tablas en la base de datos.

- Métricas de capacidad, disponible en dos formatos:
 - Archivos csv generados por el Monitor de la Java Virtual Machine implementado.
 - Archivos xml generados por el Agente Software Pandora FMS.

Esto genera la necesidad de establecer un único formato estandarizado para el manejo de los datos. En atención a este requerimiento se seleccionó el formato de archivos de texto CSV (Comma-Separated Values) por los siguientes motivos:

- Es un formato sencillo de armar y de entender.
- El usuario puede leer fácilmente la información en formato CSV a través de un editor de texto o de una hoja de cálculo.
- En Python se cuenta con librerías que permiten leer y cargar de archivos CSV de forma simple (1 línea de código).

Para la carga y estandarización de los datos, se implementó la siguiente funcionalidad:

- **Carga de archivos de demanda transaccional:** permite la carga directa de archivos CSV de datos de demanda transaccional. Esta funcionalidad es útil en el caso de que se haya extraído la información por otro medio y se la desee cargar.
- **Carga de demanda transaccional desde la base de datos:** permite obtener datos de una tabla de la base de datos que contenga varios tipos de transacciones y arma la carga transaccional en formato CSV. En el caso de aplicación, Administración de Libros, se tiene una tabla denominada “*transaccion*”, en la cual se tiene todo el registro histórico de las transacciones de registro, modificación, eliminación y consulta realizadas. A fin de que la funcionalidad sea más genérica, se permite que el usuario sea quien establece las consultas SQL que se ejecutarán hacia la base de datos y con los resultados obtenidos se crea el archivo CSV.

Esta funcionalidad también contempla una transformación de filas a columnas y el manejo de valores faltantes (missing values), ya que en la base de datos las transacciones para los diferentes tipos se encuentran como filas y para el análisis se requiere tener los valores del conteo de una misma fecha y hora como columnas. Así también, es posible que para un momento dado no existan cierto tipo de transacciones; sin embargo, en el análisis se requiere transformar esa ausencia de información a un valor cero de número de transacciones. Para ejemplificar mejor esta transformación, en la *Tabla 4* se puede apreciar el resultado que se obtendría al ejecutar una consulta que contabilice los registros de la tabla “*transacciones*” agrupado por fecha y tipo.

Tabla 4. Ejemplo de datos de demanda obtenidos desde la Base de Datos

Fecha	Tipo	Número (Conteo)
2020-08-02 23:40:16	registro	6
2020-08-02 23:40:16	actualiza	1
2020-08-02 23:40:16	consulta	3
2020-08-02 23:40:16	elimina	3
2020-08-02 23:40:17	registro	3
2020-08-02 23:40:17	consulta	1
2020-08-02 23:40:17	elimina	2

Fuente: Elaboración propia

El programa implementado permite transformar los datos antes descritos al siguiente formato CSV:

```
Fecha,nro_registro,nro_actualiza,nro_consulta,nro_elimina
2020-08-02 23:40:16,6,1,3,3
2020-08-02 23:40:17,3,0,1,2
```

Como se puede apreciar de esta manera se asegura tener valores para todo tipo de transacción y se conoce toda la carga transaccional que se tuvo para una fecha y hora específica.

- **Carga de archivos de métricas de capacidad:** permite la carga directa de archivos CSV de datos de las métricas realizadas sobre el consumo de recursos. Esta funcionalidad es utilizada en el caso de aplicación para la carga de los registros de uso de memoria obtenidos por el Monitor de la JVM (Máquina Virtual Java).

Adicionalmente, esta opción del sistema permite cargar archivos generados por el Agente Software Pandora FMS en formato XML. En este caso el programa recibe un archivo comprimido con extensión “.zip” con todos los archivos XML.

- **Conversión a CSV:** para aquellos archivos zip cargados con la funcionalidad del punto anterior, se brinda la posibilidad de generar un archivo CSV el cual realiza las siguientes acciones: descomprime el contenido, lee uno a uno los archivos de monitoreo XML generados por el Agente Software Pandora FMS y arma un nuevo archivo con el mismo nombre del comprimido, pero con extensión “.csv”. La transformación por cada xml consistió en extraer la fecha y hora del atributo “timestamp” que se encuentra en la etiqueta “agent_data” y después por cada sección “module” extraer el nombre de la métrica en la etiqueta “name” y el valor en de la

etiqueta "data". Estos valores se colocan en la columna de la métrica que corresponde de acuerdo al nombre, haciendo coincidir las columnas con el resto de medidas de los otros archivos.

A modo de ejemplo, para el contenido presentado en la *Figura 10* se genera la siguiente cabecera en el archivo CSV:

```
Fecha,CPU_Load,Memory_Used,Free_Memory
```

Y el siguiente registro con los datos:

```
2020-08-03 05:22:21,4,48.92,1818
```

La cabecera aparece una sola vez para todos los xml del zip y se ubica como primera fila en el CSV y los valores de los xml son registros que aparecen después de la cabecera uno tras otro.

El resultado de este proceso es tener toda la información como archivos de texto en formato CSV (por sus siglas en inglés Comma-Separated Values) que posteriormente será utilizado por el modelo de Inteligencia Artificial.

2) Otras tareas de preparación:

El código para esta segunda etapa se genera desde el aplicativo web; sin embargo, su ejecución es externa al aplicativo ya que el código se encuentra en lenguaje Python. Se contempló de esta manera en la arquitectura, con el fin de que el usuario final tenga mayor flexibilidad en incorporar la lógica adicional que considere para la preparación de los datos.

Dentro del código generado se automatizó las siguientes tareas adicionales de preparación:

1. Carga de archivos a formato de la librería pandas.
2. Ordenar los datos por fecha.
3. Consolidar la información a nivel de minutos en lugar de segundos, esto con el fin de reducir la variabilidad tan elevada que existía al hacer por segundo.
4. Relacionar los datos de los archivos mediante la fecha y hora de los registros.
5. Identificar los datos que son entrada para el entrenamiento del modelo (X) y los que son la salida que se desea predecir (y).
6. Normalización de los datos de entrada (X) a valores entre 0 y 1
7. Conversión de unidades de medida, en aquellos casos en los que los valores eran demasiado grandes, por ejemplo, en métricas de memoria que estaban dadas en bytes, fue necesario transformar los valores a mega bytes, es decir se dividió dos veces para 1024. Esto fue necesario ya que al correr los modelos con valores muy

altos los modelos no podían entrenar de forma correcta, por ejemplo, en las redes neuronales el aprendizaje era muy bajo de época a época.

8. Extraer datos de entrenamiento y pruebas.

Con la funcionalidad antes indicada, queda lista la información para que se pueda entrenar los modelos de inteligencia artificial, lo cual se explica en el siguiente apartado.

4.4 Modelamiento de inteligencia artificial

El ámbito de la inteligencia artificial es extenso. Desde sus inicios en 1955, este término se establecía bajo la conjetura de que una computadora podría simular todo aspecto del aprendizaje y de la inteligencia (Jackson, 2019, p. xviii). Entre estos aspectos se ha considerado: el procesamiento del lenguaje natural, el procesamiento de imágenes, la búsqueda de soluciones, clasificación y regresión de información.

En el presente trabajo se ha planteado un problema específico referente a la generación de pronósticos respecto a la capacidad de la infraestructura tecnológica y para abordarlo se hace uso de una de las áreas de la inteligencia artificial denominada aprendizaje automático o machine learning como se conoce el término en inglés.

De acuerdo a Alpaydin (2010) el aprendizaje automático consiste en la programación de computadoras para optimizar un criterio de desempeño utilizando datos históricos o de ejemplo; y Russell (2018) la define como la programación de computadoras para aprender de los datos. Ambos conceptos coinciden plenamente con la solución que se desea implementar.

Dentro del aprendizaje automático se tienen varios tipos de algoritmos entre los cuales están (Russell, 2018):

- Aprendizaje Supervisado, es indispensable contar con información que indique para cada caso la solución deseada o etiqueta. Entre los usos se tiene la predicción de valores numéricos (regresión) o la clasificación de datos.
- Aprendizaje No Supervisado, se usa principalmente para agrupamiento de información. En este caso no se tienen datos previamente clasificados, por lo que se arman grupos con los datos que tienen características más similares, infiriendo así una posible etiqueta la cual no se conoce.
- Aprendizaje Semi-supervisado, estos algoritmos usan algunos datos etiquetados junto con grandes cantidades de datos no etiquetados para construir mejores clasificadores disminuyendo los costos y esfuerzo requeridos para etiquetar datos (Zhu, 2007).

- Aprendizaje por reforzamiento, en este caso el sistema debe interactuar con el entorno y realiza un conjunto de acciones dadas como respuesta a las observaciones. El sistema recibe una recompensa al evaluar el resultado de las acciones tomadas.

Por las características del objetivo que se persigue en este trabajo, el tipo de algoritmo que mejor se adapta es el supervisado, ya que gracias a las mediciones realizadas se cuenta con la información etiquetada, la cual permitirá entrenar la solución con valores ya conocidos. Dentro de los algoritmos supervisados más importantes se tiene (Russell, 2018):

- K-nears neighbors
- Linear regression
- Neural networks
- Support vector machines
- Logistic regression
- Decision trees and random forests

Considerando este contexto a continuación se realiza la selección de los algoritmos que se implementarán para la solución.

Selección de algoritmos para la implementación

Para la selección de los algoritmos a implementar, se ha tomado en cuenta de entre los antes presentados, aquellos que han sido considerados en los trabajos identificados en la sección “2.2 Aplicación actual de la IA a la gestión de la capacidad tecnológica”. No se implementa un solo algoritmo ya que dependerá de la métrica de capacidad que se quiera analizar el identificar el más idóneo. Con el fin de brindar varias alternativas al usuario, se ha seleccionado las siguientes opciones:

- **Regresión lineal**, es una herramienta estadística que permite modelar la relación que existe entre ciertas variables y algunos valores previstos de salida (Shai & Shai, 2014). Se utilizará este algoritmo para aquellos casos en donde el consumo de recursos tecnológicos se asemeja a una función lineal.
- **Random forest**, es un tipo de algoritmo de ensamblaje y consiste en una combinación de árboles predictores (Breiman, 2001, pp. 1-2). La definición de los árboles se basa en un particionamiento binario recursivo del espacio de predicción en donde el nodo raíz comprende todo el espacio de predicción y cada árbol contempla un conjunto de variables aleatorias (Cutler, Cutler, Stevens, 2011, pp. 1-3). Breiman (2001, p. 21) propone el uso de Random Forest no solo para clasificación sino también para

regresión. En este caso el valor se obtiene mediante el promedio de los valores predichos por los árboles.

Este tipo de algoritmo permitirá contar con una alternativa de proyección no lineal, a diferencia del algoritmo del punto anterior.

- **Máquinas de vector de soporte (regresión)**, de acuerdo a Pedregosa et al. (2011) las máquinas de vector de soporte son un conjunto de herramientas de aprendizaje automático que se utilizan tanto para clasificación como para regresión y detección de anomalías. Su principal ventaja es la efectividad que tiene en espacios de gran número de dimensiones. La implementación específica para regresiones se denomina Support Vector Regression. Este tipo de algoritmos tienen varias alternativas de funciones de kernel que permiten manejar tanto esquemas lineales como no lineales como por ejemplo: lineal, polinómica, radial y sigmoid (Pedregosa et al., 2011). Para el presente trabajo se seleccionó un kernel no lineal denominado Radial Basis Function (RBF) con lo cual se complementa el abanico de opciones brindadas al usuario. En la *Figura 12* se puede observar un ejemplo del comportamiento del algoritmo seleccionado utilizando datos en una dimensión.

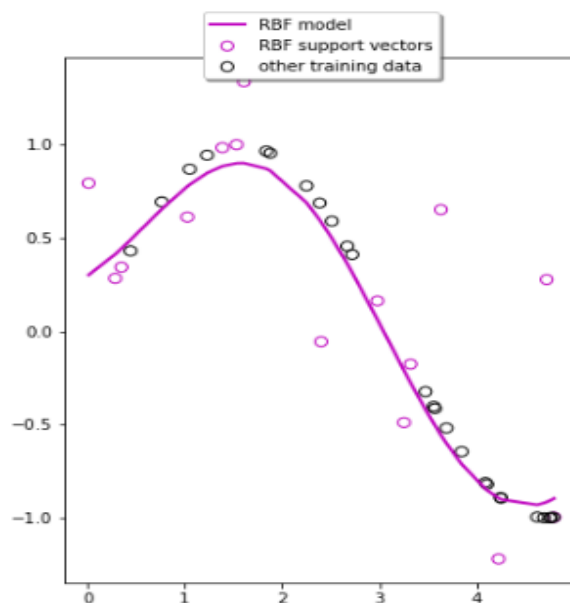


Figura 12. Ejemplo de Support Vector Regression con kernel RBF

En la imagen se puede ver el ajuste del algoritmo a predicciones no lineales

Fuente: Ejemplo disponible en: https://scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html

- **Red neuronal artificial**, son modelos computacionales inspirados por la estructura de las redes neuronales del cerebro, en donde muchas neuronas, organizadas en capas, pueden estar interconectadas y transportan el resultado de sus cálculos a las

siguientes neuronas de la red. Cada neurona toma las salidas de las neuronas anteriores (conectadas a esa neurona) y realiza las sumas de esas entradas ponderándolas por ciertos pesos. El resultado de la suma ponderada es utilizado como entrada para calcular la operación definida en la neurona como función de activación y el resultado se pasa a la siguiente neurona, a este proceso se denomina feedforward y puede ser visualizado de forma gráfica en la *Figura 13* (Shai & Shai, 2014).

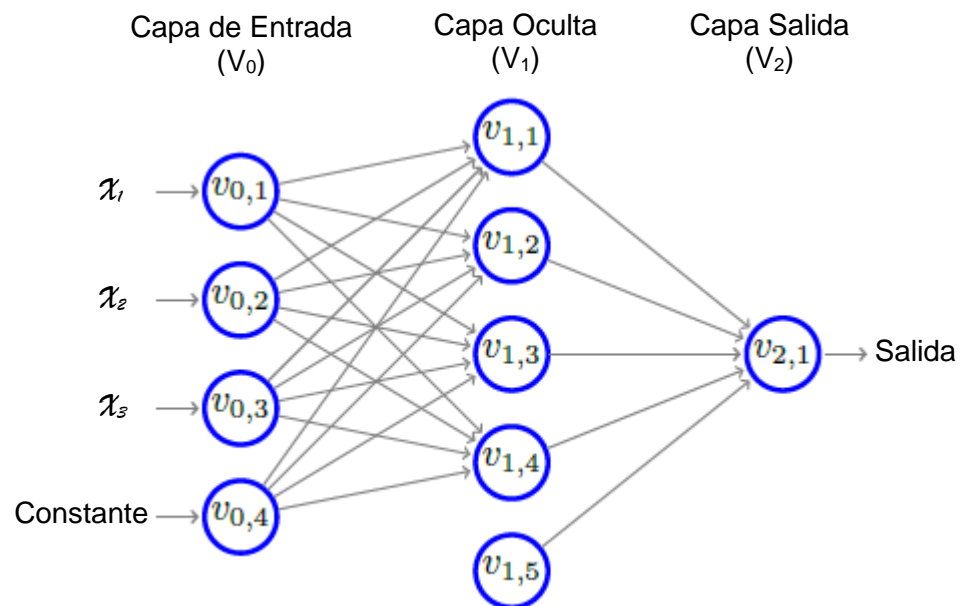


Figura 13. Red neuronal por capas y proceso feedforward

En la imagen se puede apreciar a las neuronas interconectadas en tres capas: entrada, oculta y salida. Los datos fluyen hacia adelante conforme al proceso feedforward.

Fuente: Adaptado de (Shai & Shai, 2014, p. 270)

El proceso de entrenamiento de las redes neuronales artificiales tiene por objetivo identificar los valores más adecuados de los pesos que tendrá cada entrada de cada neurona. Este proceso consiste en entregar a la red los datos de entrada para que calcule la salida y se pueda comparar estos valores con los resultados deseados. La red se ajusta en base a la retroalimentación del error que existe, buscando minimizar la diferencia entre los valores deseados y obtenidos (Basoqain, s. f., p. 17).

K-nears neighbors, son uno de los algoritmos de aprendizaje automático más simples y buscan aprender las etiquetas de nuevos datos en base a las etiquetas que tienen sus vecinos más cercanos (Shai & Shai, 2014). Estos algoritmos no se consideran por cuanto están orientados a problemas de clasificación y lo que se desea es generar una predicción numérica lo cual es un problema de regresión.

El uso de Logistic regression está orientado a tareas de clasificación en un intervalo binario (Shai & Shai, 2014). No se considera este algoritmo ya que se requieren determinar valores numéricos de proyección de uso de recursos y no están limitados a una clasificación y mucho menos a un rango binario.

Implementación de algoritmos

Los modelos fueron implementados inicialmente como cuadernos de trabajo (notebooks) utilizando lenguaje Python. Sin embargo, con el fin de tener una integración completa con el sistema desarrollado IACT se implementó en Java la generación de los archivos Notebook (.ipynb) obteniéndose la arquitectura integral del sistema de la *Figura 14*.

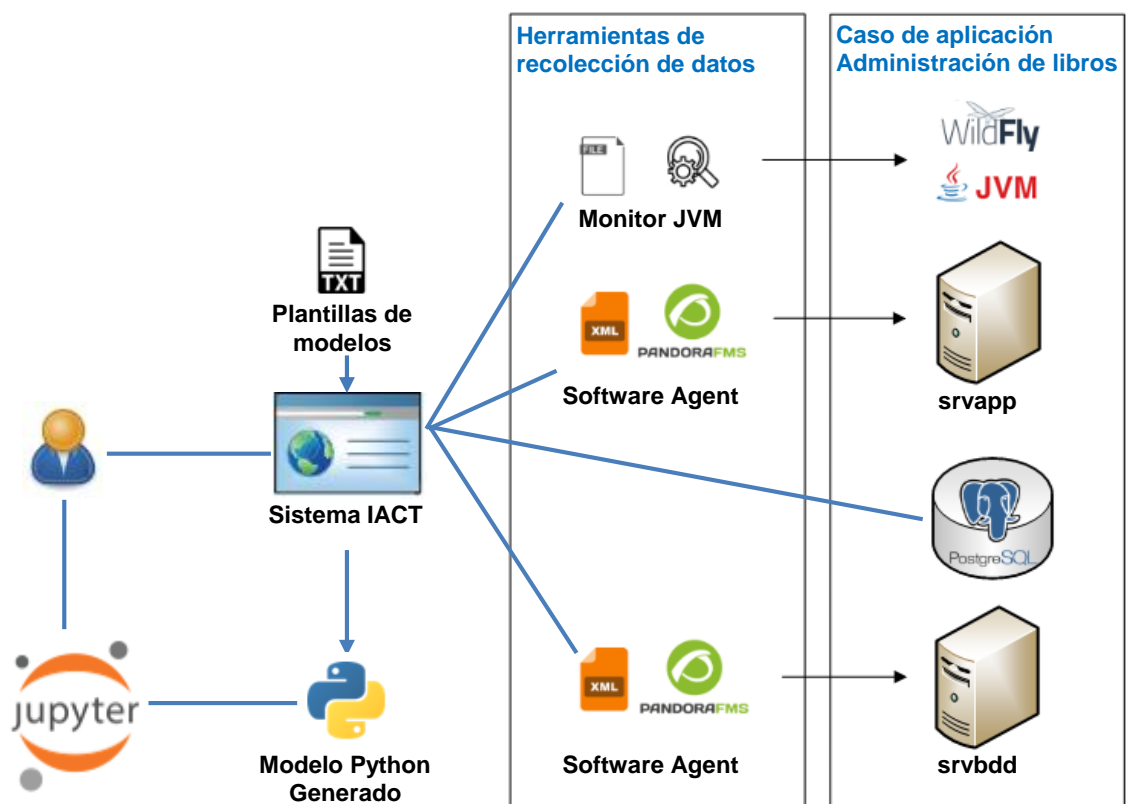


Figura 14. *Arquitectura integral de la solución IACT*

Esta figura se muestra el panorama completo de componentes desarrollados e incorporados para la solución de inteligencia artificial aplicada a la gestión de la capacidad y su interacción con el caso de aplicación (administrador de libros).

Fuente: Elaboración propia

Para la generación de los modelos en el sistema IACT se creó una clase utilitaria denominada Notebook.java la cual facilita la creación de archivos con formato notebook (.ipynb) y la incorporación tanto de comentarios (markdown) como de código (python). Este componente permite incluir contenidos python de plantillas que se encuentran en otros archivos, esto

brinda mayor flexibilidad a la hora de realizar ajustes a los modelos ya que se puede actualizar las plantillas, sin necesidad de modificar el código del sistema IACT.

Este código junto con el de las pantallas que permiten la interacción del usuario con el sistema están publicados en el proyecto iact conforme a lo indicado en la sección 7.1 Anexo I. Código publicado.

Una vez generado el modelo, el usuario lo accede, utiliza y ajusta a través de Jupyter Notebook. El cuaderno generado cuenta con secciones claramente definidas y comentadas lo cual facilita el entendimiento por parte del usuario.

Es importante mencionar que debido a la naturaleza diversa de los datos que se podrían tener en cada caso que se plantee de gestión de la capacidad de infraestructura tecnológica, no es posible establecer un modelo único que responda a todas las necesidades. Por lo tanto, los modelos que se implementaron son una referencia y deberán ser ajustados a cada caso. Los parámetros establecidos se definieron en base a los datos recolectados del caso de aplicación de Administración de libros.

Todos los modelos toman como insumo los archivos de datos CSV generados en la sección anterior 4.3 Preparación de datos. A continuación, se describen a mayor detalle las características técnicas de los algoritmos implementados:

- **Regresión lineal**, se utilizó la implementación de la librería scikit-learn con los parámetros por defecto. Este algoritmo no requiere mayor configuración.
- **Random forest**, se implementó una función que permite visualizar la evolución del error generado por el algoritmo en base a la profundidad del árbol. De esta manera el usuario final, puede identificar de mejor manera la mejor profundidad para el modelo.

La implementación realizada de este algoritmo contempla como valores: una profundidad de 5 niveles y un total de 50 árboles para las predicciones. Se configura también una semilla a través del parámetro `random_state` con el fin de que sus resultados sean reproducibles. Adicionalmente, se utiliza el error cuadrado medio “mse” (Mean Square Error) para estimar la calidad de las divisiones. El usuario puede ajustar estos valores de así considerarlo necesario.

- **Máquinas de vector de soporte (regresión)**, como se había indicado antes, este algoritmo se implementó con el kernel no lineal “rbf” (Radial Basis Function). Se utilizó también el valor de 0.8 como parámetro de regularización, el cual de acuerdo a Pedregosa et al. (2011) utiliza el cuadrado de la penalización L2. Se usa el epsilon por

defecto que es de 0.1, el cual genera un margen en el que la diferencia entre el valor predicho y el esperado no se penaliza (Pedregosa et al., 2011). Finalmente, para el parámetro gamma se consideró el uso de la opción “scale” la cual considera el número de características y la varianza que existe en los valores.

- **Red neuronal artificial**, en este caso se experimentó con varias redes neuronales y se estableció un modelo propuesto con la estructura de la *Figura 15*:

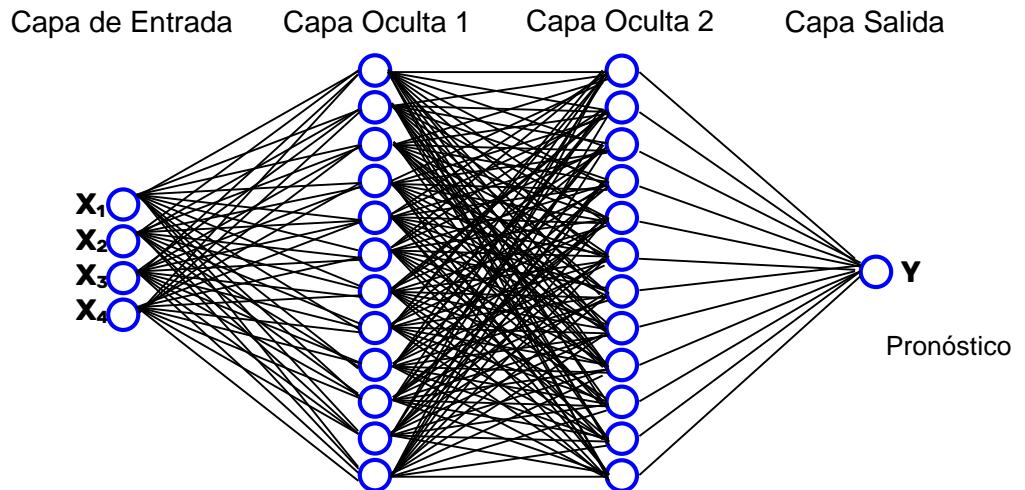


Figura 15. *Red neuronal implementada para pronósticos de capacidad tecnológica*
 En la imagen se visualiza el diseño de la red neuronal implementada como parte del sistema IACT para generar pronósticos de capacidad tecnológica.
Fuente: Elaboración Propia

Para la función de optimización se contempló el algoritmo Adam con sus parámetros por defecto y como función de pérdida se utiliza el error cuadrático medio “mse”. Adicionalmente se usa el 20% de los datos de entrenamiento para la validación del modelo. Las características de cada capa se explican a mayor detalle en la *Tabla 5*.

Tabla 5. Descripción de la red neuronal implementada

Capa de Entrada	Capa Oculta 1	Capa Oculta 2	Capa Salida
Neuronas: número dinámico en base a atributos de la demanda. Para el caso de aplicación (Administración de libros) 4.	Capa Densa Neuronas:12 Activación: ReLU Inicializador: Glorot Uniform	Capa Densa Neuronas:12 Activación: ReLU Inicializador: Glorot Uniform	Capa Densa Neuronas:1 Activación: No se coloca activador ya que se está realizando una regresión y se busca un valor de predicción abierto.

Fuente: Elaboración propia

Como se explica en la *Tabla 5*, en la capa de entrada de la red, el sistema genera el número de neuronas de forma dinámica, contando la cantidad de atributos que provienen de los archivos de demanda. Así, por ejemplo, en el caso de aplicación (Administración de libros) se contempló 4 tipos diferentes de transacciones (registro, actualización, consulta y eliminación) y por cada uno se tiene el número de transacciones ejecutadas, por esta razón en el modelo generado se tienen 4 neuronas de entrada.

En las capas ocultas, opcionalmente el usuario podría incluir dropout, la cual es una técnica de la regularización en donde de forma aleatoria se elimina un porcentaje de las neuronas definidas durante el entrenamiento, esto permite que el modelo pueda manejar los casos en los que haya sobreajuste.

A más del modelo se implementó la generación de un gráfico que permite visualizar la comparación del error cuadrático medio (mse) de los datos de entrenamiento con el mse de los datos de validación, esto permite al usuario identificar los casos en que existe sobreajuste.

Finalmente, el código contempla una evaluación de los resultados de todos los algoritmos a fin de que el usuario pueda escoger el modelo que tenga menor error. Estas métricas se explican en la siguiente sección.

4.5 Evaluación – desempeño de los modelos

Para la evaluación del modelo se ha considerado principalmente la métrica del error cuadrático medio o mean square error (MSE) como se lo conoce en inglés. Este indicador se define como la suma del cuadrado de la diferencia entre la calificación prevista y la calificación real (Dua, Ghotra, Pentreath, 2017). Esta función se utilizó tanto durante el entrenamiento

como objetivo de minimización como en una tabla comparativa que se generó de todos los modelos.

Adicionalmente, también se utilizó la raíz cuadrada del MSE, indicador denominado RMSE (Root Mean Square Error), por cuanto como indican Dua, Ghotra, Pentreath (2017) es un valor más interpretable ya que se encuentra en las mismas medidas que los datos originales.

Finalmente, se obtuvo la métrica del error absoluto medio o mean absolute error (MAE) con la cual en lugar de tomar el cuadrado se calcula el valor absoluto de la diferencia del valor real y del predicho. Estos valores también se encuentran en las mismas dimensiones de los datos.

Los algoritmos de predicción implementados toman todas las características de la demanda y predicen los valores de una métrica a la vez. Por esta razón, la evaluación se debe realizar también por cada una de las métricas de capacidad tecnológica que se van a predecir. En cada caso pueden existir comportamientos y resultados diferentes a partir de lo cual el usuario podrá ajustar los modelos propuestos.

Como una muestra del análisis realizado, a continuación, se presentan los resultados obtenidos para la generación de pronósticos de la métrica de capacidad tecnológica denominada “MemoriaUsadaJVM”, esta métrica representa la cantidad de memoria que se consume dentro del servidor de aplicaciones en el cual está instalada la aplicación web de administración de libros.

Los valores de esta métrica inicialmente se encontraban en bytes; sin embargo, como se indicó en la sección “4.3 Preparación de datos”, se identificó que para un mejor desempeño de los modelos se debía reducir el tamaño de los valores grandes, razón por la cual se transformó la métrica a mega bytes. A continuación, en la *Tabla 6*, se puede ver información relevante de la métrica, misma que fue obtenida del análisis que se encuentra en el Notebook generado por el sistema IACT.

Tabla 6. Análisis descriptivo de valores de la métrica “MemoriaUsadaJVM”

Medida	Valor (Mega Bytes)
Total de registros (count)	427,00
Media (mean)	138,17
Desviación estándar (std)	48,39
Mínimo (min)	64,04
Percentil 25%	96,82
Percentil 50%	129,71
Percentil 75%	180,22

Máximo (max)	247,64
--------------	--------

Fuente: Elaboración propia, a partir de valores presentados por describe de pandas en el notebook de los modelos

Estos datos se utilizan para dar un contexto de los resultados y permitirán completar la evaluación de resultados.

Antes de pasar a revisar las métricas comparativas, es importante analizar los resultados de las funciones implementadas para conocer el comportamiento de los modelos Random Forest y de la Red Neuronal durante el entrenamiento.

En el caso de Random Forest, se analiza la evolución que existió del error cuadrático medio respecto al nivel de profundidad y como se puede apreciar en la *Figura 16*, en lugar de disminuir el error se identifica que a mayor profundidad el error se incrementa. Por lo antes indicado, en este caso, se ajusta la profundidad del algoritmo a 2 niveles.

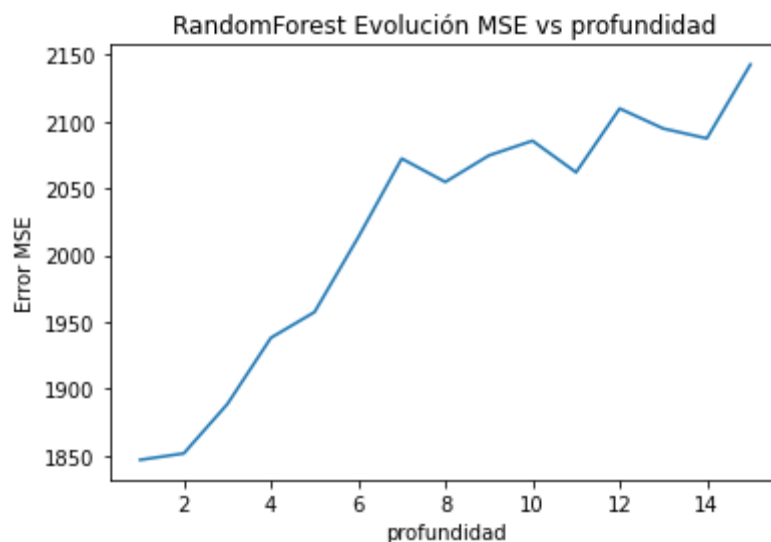


Figura 16. Análisis del error cuadrático medio respecto a profundidad de Random forest

En la imagen se puede identificar que el error se incrementa cuando se tiene mayor profundidad en el algoritmo. Con este análisis, para la predicción de memoria utilizada en el servidor de aplicaciones, se opta por manejar una profundidad de 2 en Random Forest.

Fuente: Grafico de capturado en Jupyter Notebook

En el caso de la red neuronal, se analiza la evolución del error cuadrático medio con los datos de entrenamiento y de validación a fin de identificar si existe un posible sobre ajuste en el modelo. El resultado obtenido puede visualizarse en la *Figura 17*.

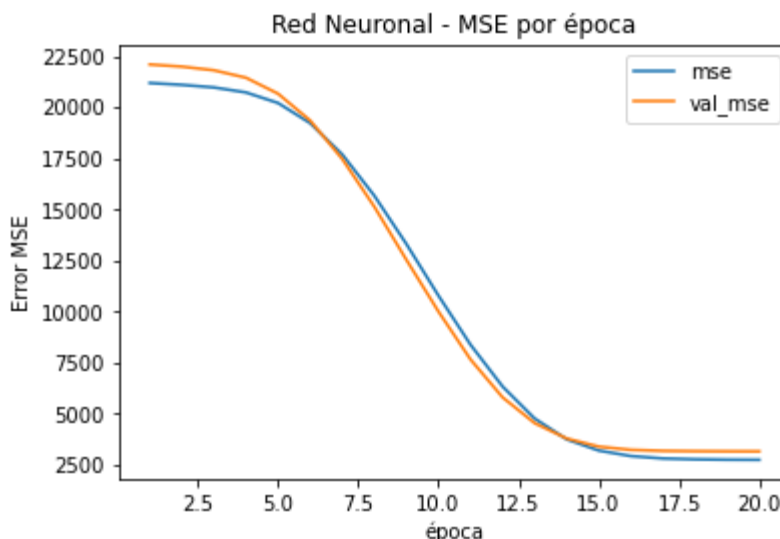


Figura 17. Análisis del error cuadrático medio por época en red neuronal

Como se puede apreciar la red neuronal tiene un comportamiento adecuado ya que por cada época se reduce el error cuadrático medio y el comportamiento es similar en los datos de entrenamiento y de validación.

Fuente: Grafico de capturado en Jupyter Notebook

En color celeste se encuentra la línea que refleja el valor del error cuadrático medio (mse) con los datos de entrenamiento para cada época y puede notarse que se reduce considerablemente hasta la época número 15, a partir de ella la reducción del error es menor y entre las épocas 17 y 20 el valor del mse se mantiene prácticamente constante. Se debe destacar también que al comparar el comportamiento entre los datos de entrenamiento (mse) y los datos de validación (val_mse), no existe mayor diferencia, esto es positivo por cuanto indica que el modelo no se encuentra sobre ajustado y que responde tanto a predicciones de datos entrenados como a nuevos datos con niveles similares de mse.

Una vez revisado el comportamiento del entrenamiento, amerita comparar el resultado de los cuatro modelos implementados, para lo cual en la *Tabla 7* se presentan, para cada algoritmo, los resultados de las tres métricas de error utilizadas.

Tabla 7. Cuadro comparativo de resultados de errores por modelo

Métrica \ Algoritmo	Regresión Lineal	Random Forest	Máquinas de Vector de Soporte (Regresión)	Red Neuronal
Error Cuadrático Medio (MSE)	1888,39	1851,72	1976,69	2371,68
Raíz de Error Cuadrático Medio (RMSE)	43,46	43,03	44,46	48,69
Error Absoluto Medio (MAE)	36,60	36,17	35,85	39,54

Fuente: Elaboración propia, a partir de valores presentados en el notebook de los modelos

En base a los resultados obtenidos en el error cuadrático medio, se puede identificar que el menor valor se encuentra para el algoritmo Random Forest; sin embargo, los otros algoritmos también presentan resultados no muy distantes. De los cuatro algoritmos el que tiene un peor resultado es la red neuronal.

Al analizar los errores en base al Error Absoluto Medio (MAE por sus siglas en inglés), quien presenta mejores resultados son las Máquinas de Vector de Soporte (Regresión) seguido por Random Forest. Considerando que el MSE penaliza en mayor medida los errores se opta para este caso por el estimador Random Forest.

Si se compara la raíz de error cuadrático medio (RMSE) de Random Forest con la información del análisis descriptivo de la métrica se puede identificar los resultados de la *Tabla 8*.

Tabla 8. Comparación de RMSE de mejor modelo con análisis descriptivo de la métrica “MemoriaUsadaJVM”

Medida	Valor (Mega Bytes)	RMSE (Mega Bytes)	Comentario
Media (mean)	138,17	43,03	El error respecto a la media representa el 31,14%
Desviación estándar (std)	48,39	43,03	El error del modelo es menor a una desviación estándar
Mínimo (min)	64,04	43,03	El error respecto al valor mínimo representa el 67.19%
Percentil 25%	96,82	43,03	El error representa el 44.44% para el primer cuartil
Percentil 50%	129,71	43,03	El error representa el 44.44% para el primer cuartil
Percentil 75%	180,22	43,03	Representa el 23.87% para el tercer cuartil
Máximo (max)	247,64	43,03	El error respecto al valor máximo representa el 17.37%

Fuente: Elaboración propia, a partir de valores presentados por describe de pandas en el notebook de los modelos

En base al análisis antes realizado tomando en cuenta el valor máximo de la métrica, el error obtenido por el mejor modelo 17.37% es menor al 20%, razón por la cual se cumple el objetivo general establecido en el proyecto. Es importante mencionar que el sistema desarrollado es una herramienta que facilita la gestión de la capacidad; sin embargo, es necesaria la intervención del usuario para afinar los modelos de manera que estos se ajusten a la información particular de cada métrica.

Como último punto se tiene la implementación de la interfaz de usuario a través de la cual el usuario interactuará con el sistema inteligente desarrollado.

4.6 Interfaces de usuario e instalación de la solución

Finalizando con la última etapa de la metodología, ver sección “3.3 Metodología CRISP-DM”, se tiene la *implementación*, la cual consiste en generar las facilidades necesarias (interfaces de usuario e instalación) para que el usuario pueda utilizar los modelos de información. En esta sección se describen estos aspectos.

Interfaces de usuario:

En la solución planteada el usuario interactúa directamente con dos entornos, por un lado, accede a las pantallas de la aplicación web java del sistema IACT y por otro lado accede a los modelos python a través de Jupyter Notebook. En la *Figura 18* se pueden apreciar las tecnologías utilizadas para la implementación de las interfaces de usuario:

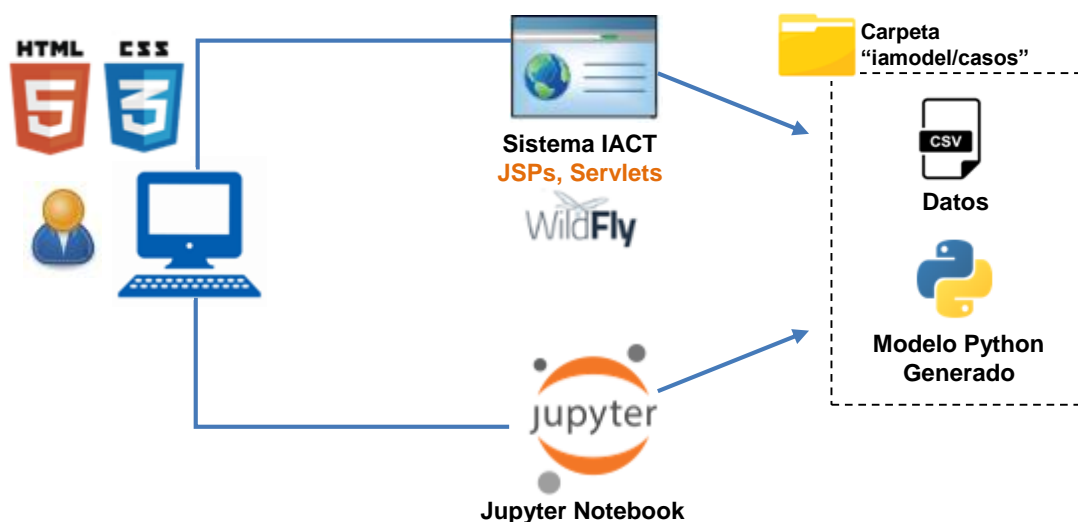


Figura 18. *Tecnologías utilizadas para la implementación de la interfaz de usuario*
En la imagen se aprecian las tecnologías de los tres ámbitos principales: el cliente (navegador), el servidor WildFly y el servidor Jupyter.

Fuente: Elaboración propia

Adicional a html 5 y css 3 a nivel de cliente se utilizan las siguientes librerías (JavaScript / css)

- Bootstrap (<https://getbootstrap.com/>), con el fin de que las pantallas tengan diseño responsivo, es decir que se adapten al tamaño de pantalla que se disponga.
- JQuery (<https://jquery.com/>), para llamadas asincrónicas al servidor. Requisito de las otras librerías.
- Dropzone.js (<https://www.dropzonejs.com/>), para carga de archivos arrastrando y soltando (drag & drop)

A nivel del servidor se utilizó:

- Java Server Pages, para generar el html dinámico de las pantallas.
- Servlets, para la lógica funcional de la aplicación. En estos componentes se incluye la preparación de los datos y la generación de los modelos en formato Notebook.

Como se puede apreciar en la *Figura 18*, se tiene una carpeta mediante la cual se comparte información entre el servidor WildFly y el Notebook. Principalmente lo que comparten son los archivos de datos y el código python que genera el sistema IACT para que el usuario pueda utilizarlo desde Jupyter.

Las pantallas del sistema web java tienen las siguientes secciones indicadas en la *Figura 19*:

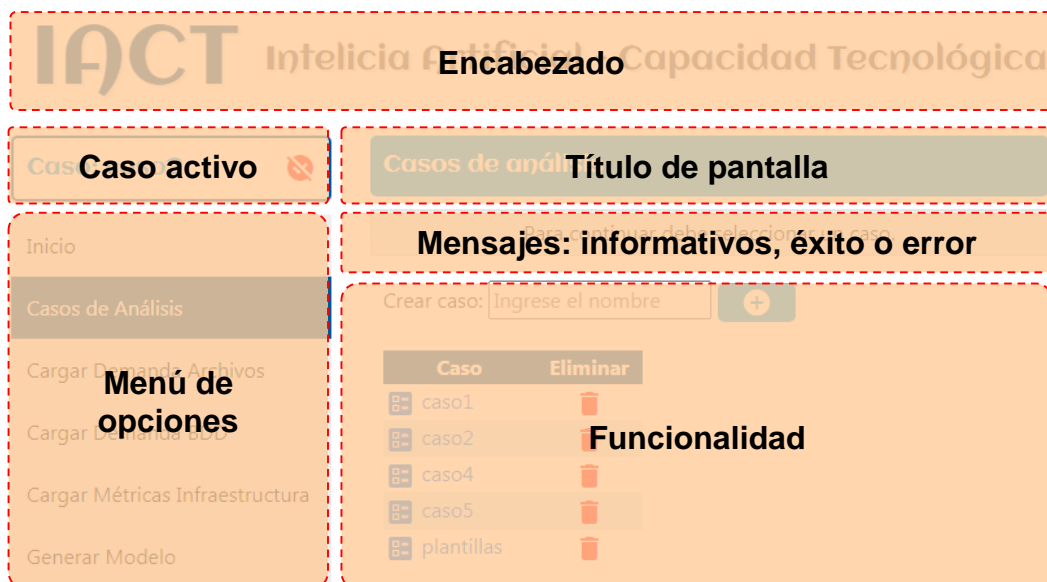


Figura 19. Estructura de pantallas del sistema IACT

En la imagen se aprecian las secciones que tiene cada pantalla del sistema de inteligencia artificial para la gestión de capacidad.

Fuente: Elaboración propia

El primer paso que debe realizar el usuario, es la creación y selección de casos, el cual lo puede realizar mediante la pantalla de la *Figura 20*:



Figura 20. Pantalla para gestión de casos del sistema IACT

Se presenta la funcionalidad para la creación, selección y eliminación de casos. Cada caso maneja sus datos y los modelos se generan para esos datos

Fuente: Captura del sistema desarrollado

Una vez seleccionado un caso, el usuario puede realizar la carga de información del demanda mediante dos opciones, la carga por archivos o la extracción de información desde la base de datos como se puede apreciar en la *Figura 21*.



Figura 21. Pantallas para gestión de datos de demanda

El usuario puede cargar información de demanda a través de archivos CSV o extrayendo la información de la base de datos

Fuente: Capturas del sistema desarrollado

El paso siguiente es la carga de las métricas de capacidad de la infraestructura tecnológica utilizada por el sistema que se está monitorizando. Como se puede observar en la *Figura 22* el usuario puede cargar información directamente en formato CSV o en su defecto archivos ZIP que contengan métricas xml generadas por Pandora Software Agent. Para estos últimos el sistema también le permite transformar su contenido al formato estándar CSV manejado en los modelos inteligentes.

Carga de métricas de infraestructura

Carga de archivos de datos

Arrastre y suelte aquí los archivos a cargar

Cargar archivos

Datos cargados

Archivo	Generar CSV	Fecha	Tamaño (Bytes)	Eliminar
monitoreo_memoria.csv		27/08/2020 10:38:52	75856	
srvapp.csv		13/09/2020 10:48:11	81223	
srvapp.zip	<input checked="" type="checkbox"/>	27/08/2020 10:39:35	1713398	
srvbdd.csv		13/09/2020 10:48:46	96736	
srvbdd.zip	<input checked="" type="checkbox"/>	30/08/2020 02:17:46	1568699	

Figura 22. Pantallas para gestión de datos de métricas de infraestructura

El usuario puede cargar información de demanda a través de archivos CSV o extrayendo la información de la base de datos

Fuente: Capturas del sistema desarrollado

Finalmente, el usuario tiene la posibilidad de generar el modelo (Notebook) a través de un wizard de tres pasos, en donde primero escoge los archivos de datos que desea analizar, después selecciona los campos a utilizar en el modelo y por último escoge los contenidos que se incluirán en la generación, así, por ejemplo: el análisis de datos y los algoritmos que se quiere evaluar para su uso. Esta secuencia se aprecia en la *Figura 23*.

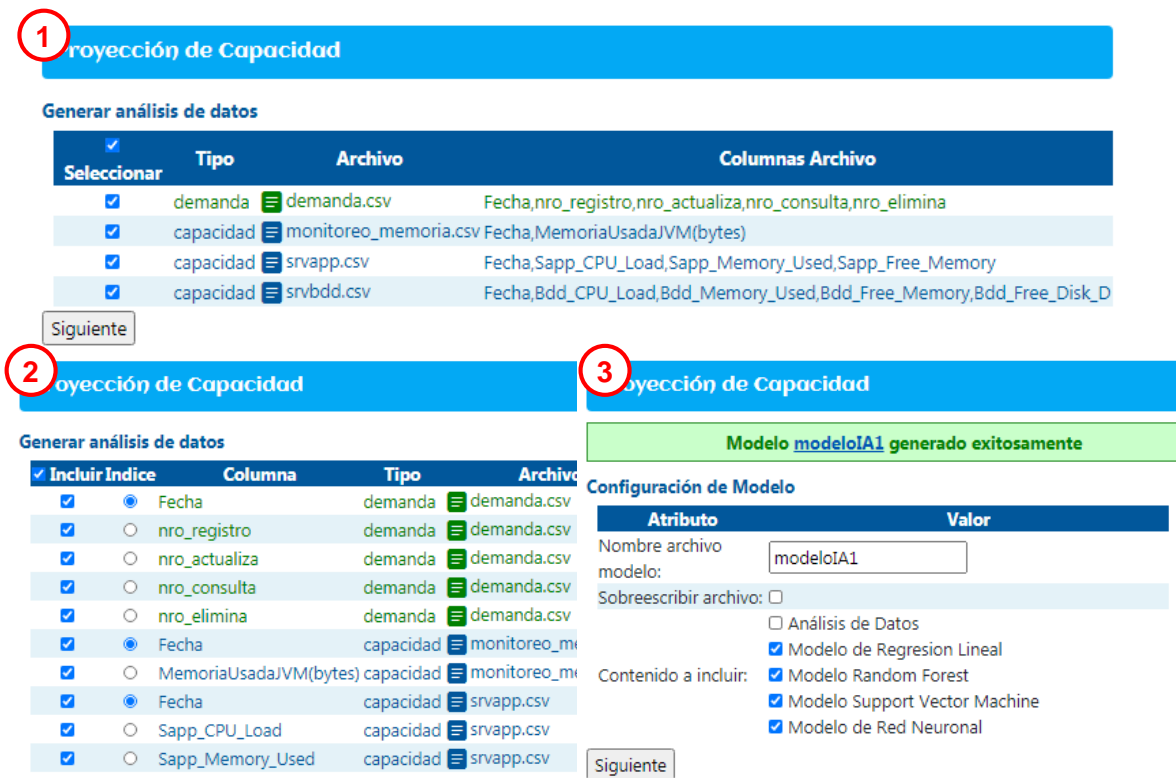


Figura 23. Pantallas para la generación del modelo inteligente

En la imagen se visualiza cómo el usuario escoge los archivos, los campos de los archivos y el contenido que se incluirá en el modelo generado.

Fuente: Capturas del sistema desarrollado

Como se aprecia en el paso 3 de la *Figura 23*, después de generar el modelo se presenta un mensaje de éxito, el cual genera un enlace directo al modelo y al hacer clic en el nombre se abre automáticamente el modelo con Jupyter Notebook. Una vez en este entorno, se puede visualizar, ejecutar y ajustar el modelo generado como se puede apreciar en la *Figura 24*.

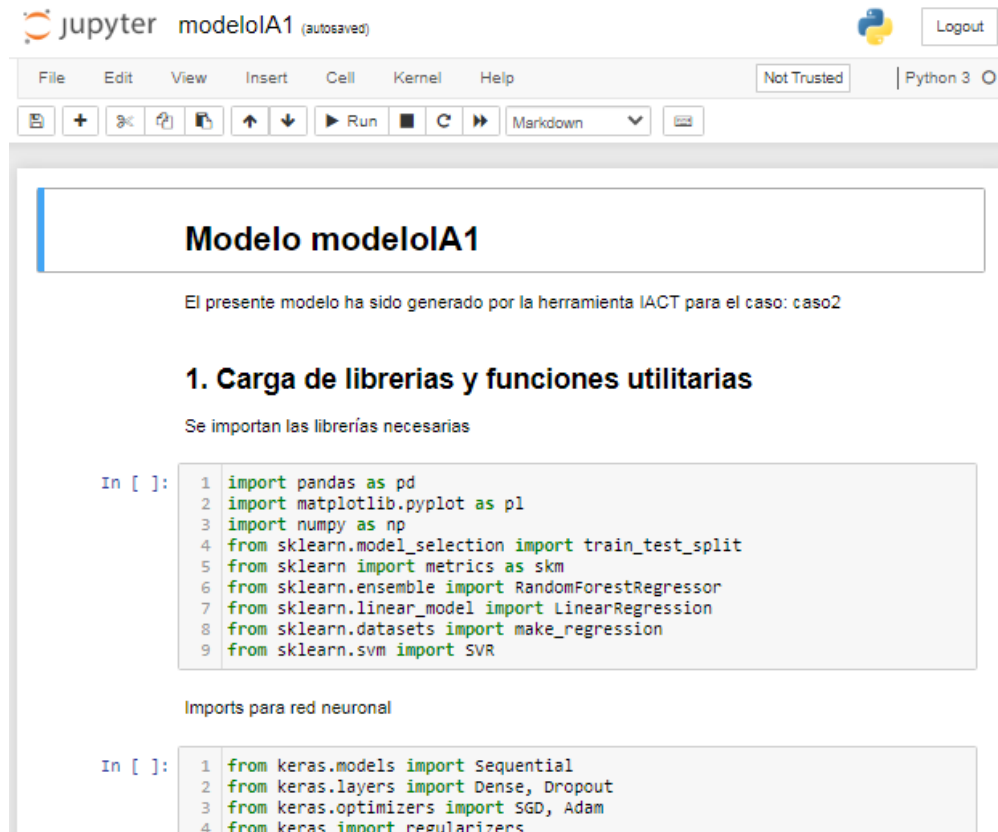


Figura 24. Modelo generado abierto en Jupyter Notebook

En la imagen se visualiza el modelo que se abre al hacer clic en el link del mensaje de éxito de la generación.

Fuente: Captura de pantalla en Jupyter Notebook

Es importante mencionar que el sistema desarrollado ha sido revisado y probado por tres usuarios expertos, fruto de lo cual se afinaron las funcionalidades que provee el sistema. Como ya fue mencionado anteriormente, esta es una herramienta de apoyo que facilita el trabajo de estimación de recursos tecnológicos; sin embargo, es indispensable que el usuario afine los modelos propuestos de acuerdo a los datos y casos particulares que vaya a tratar.

Para concluir, en el siguiente apartado se explica el detalle para instalar y levantar la solución desarrollada.

Detalle de instalación y habilitación del sistema:

Para la instalación de la solución se tienen los siguientes requisitos:

- Instalación del Java Development Kit (JDK) 1.8, se sugiere revisar la siguiente ruta: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>.
- Instalación del Servidor de Aplicaciones WildFly 18, se sugiere revisar la siguiente ruta: <https://www.wildfly.org/>

- Instalación de ambiente Python 3.8.x, con las librerías siguientes:
 - Numpy
 - Pandas
 - Matplotlib
 - Scikit-learn
 - Tensorflow
 - Keras
 - Jupyter

Se sugiere utilizar ambientes virtuales, para evitar problemas de compatibilidad de librerías y considerar la guía de la siguiente ruta <https://www.pyimagesearch.com/2016/11/14/installing-keras-with-tensorflow-backend/>

Una vez que se tienen disponibles los prerequisites se debe realizar las siguientes actividades:

- Instalar la aplicación IACT en el servidor de aplicaciones (WildFly), para lo cual se debe:
 - Copiar el archivo “iact.war” que se encuentra en la carpeta “target” del código del proyecto iact (ver 7.1 Anexo I. Código publicado) en la carpeta “standalone/deployments” del servidor de aplicaciones WildFly.
 - Copiar la carpeta “iamodel” que se encuentra en la carpeta “target” a la misma altura que se encuentra la carpeta del servidor WildFly.
- Levantar el servidor ejecutando el comando “standalone.bat -b 0.0.0.0” en la carpeta “bin” de WildFly.
- Levantar el servidor Jupyter Notebook ejecutando el comando “jupyter notebook” ubicándose dentro de la carpeta “iamodel/casos”.
- Ingresar a la ruta: <http://localhost:8080/iact>

5 Conclusiones y trabajo futuro

5.1 Conclusiones

Existe un incremento de inversión en infraestructura tecnológica por parte de las empresas que buscan asegurar su rentabilidad (Expansión, 2019) y de igual manera las organizaciones del sector público han adoptado la tecnología para apalancar el desarrollo de un gobierno electrónico (Naciones Unidas, 2020, p. XXIV-XXV). En este contexto, es un factor de vital importancia, que las organizaciones puedan identificar de manera fácil y oportuna la capacidad de la infraestructura tecnológica necesaria para soportar su operación actual y futura. Este análisis conlleva gran complejidad y dificultad ya que requiere de conocimientos profundos de tecnología y del negocio. Adicionalmente, se requieren de herramientas que permitan establecer la relación entre la demanda del negocio y el uso de los recursos tecnológicos.

Un análisis incorrecto de la capacidad tecnológica puede ocasionar sobredimensionamiento o subdimensionamiento. Mientras el primero afecta a la eficiencia en el uso de recursos y genera costos innecesarios, el segundo pone en riesgo la operación de la organización (Office of Government Commerce, 2007c, p. 129).

El trabajo realizado permitió construir una herramienta de software que haciendo uso de técnicas de aprendizaje automático, facilita el análisis de la capacidad de la infraestructura tecnológica.

Para abordar el problema se exploró las mejores prácticas de gestión de la capacidad y se identificó la fuerte relación que existe con la gestión de la demanda. También se realizó una búsqueda exhaustiva de investigaciones relacionadas a la aplicación de la inteligencia artificial a la gestión de capacidad tecnológica y se pudo notar que es un campo poco explotado. Los trabajos identificados tienen una orientación diferente, mientras algunos contemplan la administración de nubes públicas y otros tratan de infraestructura de telefonía o eléctrica, ninguno aborda la identificación de capacidad de infraestructura tecnológica considerando su relación con la demanda transaccional.

Para el desarrollo se utilizó la metodología CRISP-DM, la cual parte del entendimiento del negocio, continúa con el entendimiento y preparación de los datos y con esta información se desarrolla el modelamiento, se lo evalúa y finalmente se implementa las facilidades para que el usuario final pueda hacer uso de los modelos creados.

Para el entendimiento de negocio, se definió un caso concreto de aplicación denominado Administración de Libros. Para el cual se construyó un sistema web y un cliente que simula la carga transaccional. Con el caso definido, se levantaron y establecieron los siguientes requerimientos de la solución: gestión de casos de análisis, carga de datos de demanda, carga de métricas de capacidad y la generación y uso del modelo inteligente. Después, se plantearon e implementaron los esquemas de monitoreo de la infraestructura y al correr la simulación de carga, se tomaron las métricas de uso de recursos como: memoria, procesamiento y almacenamiento.

A continuación, se analizó la información recolectada y se implementó la preparación de los datos en dos fases, la primera para estandarizar la información a formato de archivos CSV (valores separados por comas) y la segunda que consolidó y preparó los datos para los modelos inteligentes.

Con los datos listos, se seleccionaron e implementaron los siguientes modelos de aprendizaje supervisado: Regresión Lineal, Random Forest, Máquinas de Vector de Soporte (Regresión) y Red Neuronal Artificial. Después se corrió los modelos para la medida de uso de memoria JVM del caso de administración de libros, y se analizó los resultados utilizando tres métricas de calidad: error cuadrático medio, raíz del error cuadrático medio y error absoluto medio. Como resultado se obtuvo que el modelo que presentaba el menor error cuadrático medio fue Random Forest (1851,72), seguido por la Regresión Lineal (1888,39) y las máquinas de vector de soporte con regresión (1976,69), el mayor error se tuvo en la Red Neuronal (2371,68). Por otro lado, el algoritmo que presentó el menor error absoluto medio fue máquinas de vector de soporte con regresión (35,85 MB), seguido por Random Forest (36,17) y Regresión lineal (36.6). De igual manera el de mayor error fue la red neuronal con (39,54).

Al comparar la raíz del error cuadrático medio de Random Forest (43,03 MB), con el mayor valor de la métrica (247,64 MB) se obtuvo un porcentaje de 17.37%, cumpliendo así, el objetivo general de la investigación que era, crear una herramienta inteligente que permita proyectar los requerimientos de capacidad con un error menor al 20% del valor máximo de la métrica.

El sistema implementado IACT (Inteligencia Artificial para Capacidad Tecnológica) fue probado por tres usuarios expertos y cuenta con interfaces de usuario robustas y con diseños responsivos e intuitivos. Durante las pruebas, se confirmó que el sistema facilita la carga y la preparación de los datos de demanda y de métricas de uso de recursos de infraestructura. Adicionalmente, IACT es capaz de generar un modelo completo en formato Notebook que

incluye: el análisis, la preparación de datos, los modelos y la evaluación de resultados. Dicho modelo es accesible al usuario para su afinamiento y uso mediante Jupyter Notebook.

5.2 Líneas de trabajo futuro

Una vez que ya se ha confirmado la validez de la solución, en base a la información del caso de aplicación, se plantea que esta herramienta sea instalada y utilizada para analizar datos en un entorno real, el cual podría incluir métricas de componentes de infraestructura adicionales. En base a la retroalimentación que se obtenga del uso del sistema en el entorno real, se podría ajustar los modelos propuestos o se podría plantear la generación de modelos adicionales que permitan expandir y mejorar las capacidades predictivas de la solución.

El esquema de plantillas y de generación de modelos predictivos que se implementó en el aplicativo web java, puede servir de referencia para otras iniciativas de inteligencia artificial, en las cuales se requiera integrar ambientes inteligentes de análisis de información construidos en python con aplicaciones web desarrolladas en otros lenguajes.

El sistema implementado para el caso de aplicación puede ser utilizado para generar otros conjuntos de datos que permitan realizar mayor investigación de Inteligencia Artificial aplicada a la gestión de capacidad tecnológica.

6 Bibliografía

- Alpaydin, E. (2010). *Introduction to machine learning* (2.^a ed.). MIT Press.
- Basoqain, X. (s. f.). *Redes Neuronales Artificiales y sus Aplicaciones*.
- Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1).
- Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0 Step-by-step data mining guide*.
- Cutler, A., Cuttler, D., & Stevens, J. (2011). *Random Forests*. https://doi.org/10.1007/978-1-4419-9326-7_5
- Dekkers, R. (2003). Strategic capacity management: Meeting technological demands and performance criteria. *Journal of Materials Processing Technology*, 139(1), 385-393. [https://doi.org/10.1016/S0924-0136\(03\)00505-3](https://doi.org/10.1016/S0924-0136(03)00505-3)
- Dua, R., Ghotra, M. S., & Pentreath, N. (2017). *Machine Learning with Spark—Second Edition*. Packt Publishing.
- Eremia, M., Liu, C.-C., & Edris, A.-A. (2016). *Advanced Solutions in Power Systems: HVDC, FACTS, and Artificial Intelligence*. John Wiley & Sons.
- Expansión. (2019, octubre 25). La receta para maximizar la inversión en tecnología. *Expansión.com*. <https://www.expansion.com/economia-digital/companias/2019/10/25/5db1df50468aeba9468b46fb.html>
- Félix-Sánchez, A., & Calvo-Manzano, J. A. (2015). Comparison of models and standards for implementing IT service capacity management. *Revista Facultad de Ingeniería Universidad de Antioquia*, 74, 86-95.
- Gómez Barroso, C., & Valcarcel Martínez, L. (2019). Proceso de gestión de la capacidad y disponibilidad para un modelo de desarrollo de sistemas de servicios de tecnología de la información. *I+D Tecnológico*, 15(2), 56-67. <https://doi.org/10.33412/idt.v15.2.2234>
- Huang, J., Tang, Y., & Chen, S. (2018). Energy Demand Forecasting: Combining Cointegration Analysis and Artificial Intelligence Algorithm - ProQuest. *Mathematical Problems in Engineering*, 2018. <https://doi.org/DOI:10.1155/2018/5194810>
- ISACA. (2018). *COBIT 2019 Marco de Referencia—Objetivos de Gobierno y Gestión*.
- Jackson, P. C. (2019). *Introduction to Artificial Intelligence: Third Edition*. Courier Dover Publications.

- Jiang, Y., Perng, C.-S., Tao, L., & Rong, C. (2012, abril). *Intelligent Cloud Capacity Management*. 2012 IEEE Network Operations and Management Symposium, Maui, HI, USA. <https://doi.org/10.1109/NOMS.2012.6211941>
- Jiang, Y., Perng, C.-S., Tao, L., & Rong, C. (2013). Cloud Analytics for Capacity Planning and Instant VM Provisioning. *IEEE Transactions on Network and Service Management*, 10(3), 312-325. <https://doi.org/10.1109/TNSM.2013.051913.120278>
- Kelleher, J. D., Namee, B. M., & D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. MIT Press.
- Manasson, A. (2019, diciembre 12). *Why using CRISP-DM will make you a better Data Scientist*. Medium. <https://towardsdatascience.com/why-using-crisp-dm-will-make-you-a-better-data-scientist-66efe5b72686>
- Muñoz, F., Argente, E., Espinosa, A., Gáldamez, P., García, A., Marín, R., & Sedra, J. (2013). *Concurrencia y sistemas distribuidos*. Universidad Politécnica de Valencia. https://riunet.upv.es/bitstream/handle/10251/70991/TOC_6084_01_01.pdf?sequence=5
- Naciones Unidas. (2020). *E-Government Survey 2020 Digital Government in the Decade of Action for Sustainable Development*. [publicadministration.un.org/egovkb/Portals/egovkb/Documents/un/2020-Survey/2020%20UN%20E-Government%20Survey%20\(Full%20Report\).pdf](http://publicadministration.un.org/egovkb/Portals/egovkb/Documents/un/2020-Survey/2020%20UN%20E-Government%20Survey%20(Full%20Report).pdf)
- Office of Government Commerce. (2007a). *ITIL Service Design*. The Stationery Office.
- Office of Government Commerce. (2007b). *ITIL Service Strategy*. The Stationery Office.
- Pandora FMS Enterprise. (2015). *Pandora FMS 6.0 Guía de Administración* (12.^a ed.).
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pérez-Romero, J., Sallent, O., Ferrús, R., & Agustí, R. (2015). Artificial Intelligence-based 5G network capacity planning and operation. *2015 International Symposium on Wireless Communication Systems (ISWCS)*, 246-250. <https://doi.org/10.1109/ISWCS.2015.7454338>
- Project Gutenberg. (2020, agosto 14). *Free eBooks—Project Gutenberg*. <https://www.gutenberg.org/>

- Russell, R. (2018). *Machine Learning Step-by-step guide to implement machine learning algorithms with python*. CreateSpace Independent Publishing Platform.
- Savvides Andy, P. (2004). Capacity Management. *The Computer Bulletin*, 46(5), 28-28. <https://doi.org/10.1093/combul/46.5.28>
- Seyedeh, N. F., Deo, R. C., Shojafar, M., Conti, M., & Shamshirband, S. (2018). Computational Intelligence Approaches for Energy Load Forecasting in Smart Energy Management Grids: State of the Art, Future Challenges, and Research Directions - ProQuest. *Journal Energies*, 11(3). <https://doi.org/DOI:10.3390/en11030596>
- Shai, S.-S., & Shai, B.-D. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.
- Veit, A., Xu, Y., Zheng, R., Chakraborty, N., & Sycara, K. (2014). Demand Side Energy Management via Multiagent Coordination in Consumer Cooperatives. *Journal of Artificial Intelligence Research*, 50, 885-922. <https://doi.org/10.1613/jair.4416>
- Wirth, R., & Hipp, J. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining*. 11.
- Zhu, X. (2007). Semi-Supervised Learning Literature Survey. *University of Wisconsin – Computer Sciences - Technical Report*, 1530. <https://doi.org/10.1.1.99.9681>

7 Anexos

7.1 Anexo I. Código publicado

Proyecto	Descripción	Ubicación de código
caso-libros	<p>Aplicación web desarrollada en java que recibe peticiones de registro, modificación, consulta y eliminación lógica de libros mediante protocolo HTTP y los registra en la Base de Datos.</p> <p>Este proyecto incluye la carpeta scriptBDD en la cual se encuentran los scripts de base de datos para el borrado y creación de estructuras.</p>	https://github.com/ccevallos/caso-libros.git
cliente-caso-libros	<p>Aplicación de escritorio desarrollada en java y que genera transacciones HTTP para el aplicativo caso-libros. Este programa simula una carga cíclica y utiliza información del Proyecto Gutenberg para el registro, y modificación de datos de libros.</p>	https://github.com/ccevallos/cliente-caso-libros.git
iact	<p>Corresponde a la herramienta de inteligencia artificial que permite cargar la información de demanda transaccional y de monitoreo que se tenga en formato csv o en formato del Software agente de Pandora (xml). A partir de los datos permite generar modelos en formato notebook utilizando lenguaje python para la predicción de capacidad.</p>	https://github.com/ccevallos/iact.git
modelo-caso-libros	<p>Contiene el código del modelo generado desde la herramienta iact y con los ajustes pertinentes para el caso de prueba (Administración de Libros - MemoriaUsadaJVM)</p>	https://github.com/ccevallos/modelo-caso-libros.git

7.2 Anexo II. Artículo de investigación

Inteligencia artificial aplicada a la gestión de capacidad de infraestructura tecnológica

Carlos Xavier Cevallos Terán

Universidad Internacional de la Rioja, Logroño (España)

Fecha: 23 de septiembre de 2020



RESUMEN

Un dimensionamiento adecuado de infraestructura tecnológica puede redundar en servicios disponibles y ahorros de costos. Las investigaciones actuales en inteligencia artificial, contemplan proveedores en la nube y otros sectores. No se identifican soluciones para usuarios de infraestructura de centro de cómputo. La contribución de este trabajo es una herramienta inteligente que facilita la gestión de capacidad de infraestructura tecnológica. Se construyó la solución utilizando la metodología CRISP-DM y un caso de aplicación. El sistema contempla la gestión de datos de demanda y capacidad, y la generación de pronósticos con los modelos de aprendizaje supervisado: Regresión Lineal, Random Forest, Máquinas de Vector de Soporte y Redes Neuronales Artificiales. De acuerdo al error cuadrático medio, el mejor modelo fue Random Forest, con un error inferior al 20% respecto al máximo valor de la métrica analizada. De cara al futuro, se plantea el uso y afinamiento de esta herramienta en entornos reales.

PALABRAS CLAVE

Gestión de Capacidad, Inteligencia Artificial, Aprendizaje Supervisado

I. INTRODUCCIÓN

LA infraestructura de hardware habilita los servicios tecnológicos en todo tipo de organizaciones sean estas públicas o privadas, y por consiguiente debe estar bien dimensionada. En los últimos cuatro años, se ha identificado un incremento superior al 60% del presupuesto de innovación tecnológica tanto en empresas líderes, como en las rezagadas [1]. Por otro lado, la encuesta E-government survey 2020 señala también un crecimiento en el sector público respecto al desarrollo del gobierno electrónico [2]. Considerando que la innovación requiere de infraestructura tecnológica, es necesario contar con una herramienta que permita identificar la capacidad necesaria que debe tener dicha infraestructura para optimizar y aprovechar al máximo las inversiones.

Las consecuencias principales de una inadecuada gestión de capacidad son el sobredimensionamiento, que incrementa los costos, o el subdimensionamiento que podría afectar la disponibilidad de los servicios tecnológicos de la empresa [3].

La gestión de capacidad abarca varias actividades que van desde la revisión de la capacidad, demanda y desempeño actual hasta la estimación de necesidades de recursos de infraestructura tecnológica. Esta última tarea, es la de mayor complejidad y dificultad ya que:

- Requiere de conocimiento técnico y de negocio profundo.
- La relación entre transacciones de negocio y el uso de recursos tecnológicos no se resuelven con un simple cálculo matemático, ya que el comportamiento de la infraestructura puede variar en escenarios de alta concurrencia.

En el presente proyecto se plantea la implementación de un software que tome información histórica de uso de recursos tecnológicos y demanda transaccional, e identifique automáticamente una proyección de necesidades de capacidad de infraestructura utilizando algoritmos de aprendizaje supervisado.

Para la implementación de la solución propuesta se utilizó la metodología CRISP-DM y después de la evaluación de los modelos implementados, se tuvo un resultado favorable. El modelo con menor error cuadrático medio para la métrica analizada, "MemoriaUsadaJVM", fue Random Forest; y la raíz del error cuadrático medio fue menor al 20% del mayor valor que puede tener la métrica, cumpliéndose de esta manera el objetivo planteado.

II. ESTADO DEL ARTE

La gestión de las unidades de tecnología ha evolucionado y cada vez representa mayores retos, hoy en día, se requiere de una gestión efectiva y una operación de tecnología predecible y confiable, incorporando la eficiencia en tiempo y recursos [4]. La gestión de la capacidad de los servicios tecnológicos se enfoca justamente en el uso eficiente de los recursos y ha sido abordada a nivel procedimental desde diferentes modelos y estándares entre los cuales se encuentran las normas ISO/IEC 20000:2005, la Biblioteca de Infraestructura de Tecnologías de la Información (ITIL por sus siglas en inglés) y COBIT.

De acuerdo al marco de referencia de ITIL, el objetivo de la gestión de la capacidad es asegurar la capacidad de TI (Tecnologías de la Información) en todas sus áreas para cumplir, de forma oportuna y a un costo justificable, con las necesidades actuales y futuras de negocio [3].

Para una correcta Gestión de la Capacidad, es importante también comprender el proceso de Gestión de la Demanda, ya que este le provee de información valiosa respecto a los patrones de actividad del negocio y las necesidades que se deberán satisfacer. La gestión de la demanda, de acuerdo a lo que se indica en el libro de Estrategia del Servicio de ITIL, es un aspecto crítico y si no es bien manejada puede ser una fuente de riesgo para la provisión de servicios, ya que no se podría advertir la demanda que debe ser soportada por las TI [5]. En la Figura 1, se observa el lazo estrecho que existe entre ambos procesos. La demanda influencia

los procesos de negocio y estos exigen a la vez capacidad de los servicios tecnológicos [5]

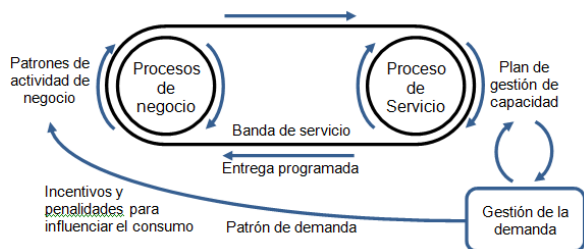


Fig. 1 Relación entre gestión de demanda y gestión de capacidad. Muestra la estrecha interacción que existe entre los procesos de negocio, la gestión de la demanda y la gestión de la capacidad. Fuente: Basado en ITIL Service Strategy [5]

En consecuencia, la capacidad requerida de la infraestructura tecnológica de una organización depende de los patrones de actividad del negocio.

Se realizó una búsqueda intensiva, en varias fuentes académicas reconocidas acerca de trabajos previos de Inteligencia Artificial aplicada a la gestión de la capacidad y/o gestión de la demanda asociada a infraestructura tecnológica, entre los trabajos identificados se tiene:

- Dos trabajos que están enfocados en la gestión de la capacidad desde una perspectiva de proveedores de infraestructura en la nube [6][7]. El presente trabajo aborda el problema como consumidores y no se limita a la nube. Al analizarse desde el lado del consumidor se cuenta con información adicional de la demanda que será utilizada en los modelos predictivos y que no fue considerada en los estudios indicados.
- Un artículo enfocado a infraestructura de operadoras de red de comunicaciones 5G [8], en este caso también se tiene la perspectiva de proveedor, mientras que el trabajo de este documento se orienta al análisis de capacidad de infraestructura tecnológica de centro de cómputo, principalmente servidores, no comunicaciones móviles 5G.
- Cuatro artículos orientados a la aplicación de la Inteligencia Artificial a la gestión de demanda. Estos casos corresponden al sector eléctrico y no contempla la gestión de capacidad de la infraestructura tecnológica de centro de cómputo [9][10][11][12]. El último, adicionalmente corresponde a un esquema multiagente, el cual es distinto al enfoque de pronósticos que es el abordado en el presente trabajo.

De los trabajos revisados se toma en cuenta el uso de algoritmos de regresión lineal, redes neuronales y máquinas de vector de soporte.

III. OBJETIVOS Y METODOLOGÍA

El objetivo general de este trabajo es el facilitar la gestión de la capacidad de la infraestructura tecnológica a través de la provisión de una herramienta inteligente que permita pronosticar los requerimientos de capacidad con un error menor al 20% del valor máximo de la métrica.

Como objetivos específicos se tiene:

1. Establecer un caso de aplicación para el análisis de la capacidad de infraestructura tecnológica.

2. Implementar el monitoreo de la infraestructura y recolectar datos.
3. Establecer los requerimientos del sistema en base al caso de aplicación.
4. Analizar los datos recolectados y prepararlos para los modelos.
5. Identificar las técnicas de aprendizaje supervisado más adecuadas e implementar los modelos de inteligencia artificial.
6. Evaluar la solución planteada con base al caso de aplicación definido.
7. Implementar la interfaz de uso del modelo.

Para la implementación de la solución de inteligencia artificial se utilizó la metodología CRISP-DM (CRoss-Industry Standard Process for Data Mining) dentro de la cual se contempla el entendimiento del negocio, el entendimiento y preparación de los datos, el modelamiento, la evaluación de resultados y la implementación de las interfaces para el usuario final como se puede ver en la Figura 2.

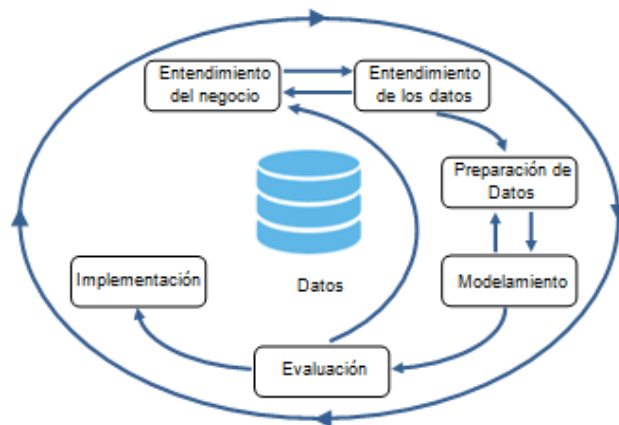


Fig. 2. Fases de la metodología CRISP-DM. Esta figura muestra las fases de la metodología seleccionada para la implementación de la solución Fuente: Basado en [13]

IV. CONTRIBUCIÓN

La solución de inteligencia artificial implementada tomó el nombre de IACT por ser una herramienta de Inteligencia Artificial aplicada a gestión de la Capacidad Tecnológica. A continuación, se describen las principales actividades ejecutadas y los resultados obtenidos en cada fase ejecutada

Entendimiento del Negocio:

En primer lugar, se definió un caso de aplicación que permite solventar los dos retos identificados:

- Existe gran diversidad de infraestructura tecnológica y se requiere acotar el problema.
- Se requiere información de demanda y de métricas de capacidad para poder entrenar los modelos, y no se han identificado bancos públicos de datos que solventen esta necesidad.

Se define como caso, un sistema de administración de información de libros y como alcance se contempla la funcionalidad de registro, actualización, consulta y eliminación

de libros. La transaccionalidad del registro de libros se simula utilizando información del Proyecto Gutenberg que tiene una biblioteca de libros gratuitos [14] y como patrón de carga su usa una variación de una función sinusoidal, que al incluir aleatoriedad genera un comportamiento similar al de la Figura 3.

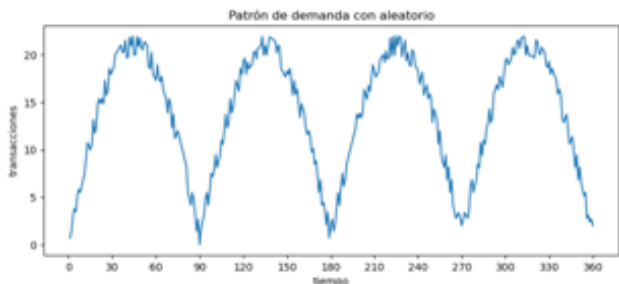


Fig. 3. Patrón de demanda con factor aleatorio para registro de libros. En esta figura se puede observar el patrón de demanda considerando un factor aleatorio. Fuente: Elaboración propia

Para el resto de transacciones se considera una carga aleatoria, entre 0 y 8 para actualizaciones, 0 y 5 para eliminaciones y 0 y 10 para consultas.

A continuación, en base al caso se levantó los requerimientos para el sistema inteligente, mismos que a nivel macro se encuentran representados en la Figura 4 como casos de uso.

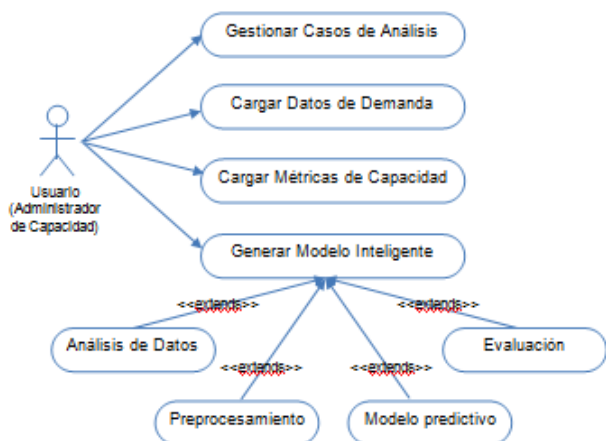


Fig. 4. Casos de uso del sistema inteligente de gestión de capacidad. En esta figura se presentan en formato UML (Unified Modelling Language) las principales funcionalidades a las cuales podrá acceder el usuario en el sistema inteligente. Fuente: Elaboración propia

Entendimiento de los datos:

Para la implementación de la solución se identifican dos tipos principales de datos:

- Demanda transaccional, esta información se encuentra en tablas de la base de datos
- Métricas de uso de recursos de infraestructura, las cuales se capturan con dos componentes.
 - Monitor de JVM (Java Virtual Machine), es un componente implementado a la medida para obtener información de la memoria dentro del JVM. Genera los datos en formato CSV.
 - Agente Software Pandora FMS, que toma métricas del computador y los registra como un conjunto de archivos XML

En la Figura 5. Se pueden identificar los monitores que se utilizan para capturar métricas de cada componente del caso de

aplicación.

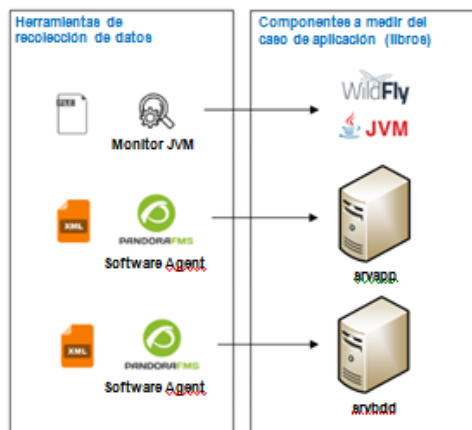


Fig. 5. Arquitectura de componentes de recolección de métricas. Esta figura muestra la arquitectura de los componentes encargados de la recolección de métricas de consumo de recursos tecnológicos para el caso de aplicación (administrador de libros). Fuente: Elaboración propia

En este análisis se identifican dos temas clave a considerar para tener datos válidos. Antes de medir se debe asegurar lo siguiente:

- Que la hora de los computadores se encuentre sincronizada.
- Que se maneje un mismo formato de fecha en todas las herramientas de monitoreo.

El no considerar esto podría invalidar la prueba.

Preparación de datos:

La preparación de datos fue concebida en el sistema IACT, Sistema de Inteligencia Artificial desarrollado para la Gestión de la Capacidad Tecnológica, en dos etapas:

- Carga de datos y estandarización, el sistema permite cargar la información y en los casos que corresponde brinda la funcionalidad de transformación a un único formato de datos.
- Otras tareas de preparación, una vez homogenizados los formatos, se implementan en python el resto de tareas como son: el ordenado y consolidación de datos, el relacionamiento de los diferentes archivos, la selección de datos de entrada y salida, la normalización de datos de entrada, y la separación de datos de entrenamiento y pruebas.

Modelamiento de inteligencia artificial:

En esta sección se analizaron algoritmos de aprendizaje supervisado con regresión y se definió la implementación de cuatro algoritmos diferentes que se listan a continuación:

1. **Regresión lineal**, es una herramienta estadística que permite modelar la relación que existe entre ciertas variables y algunos valores previstos de salida [15]. Para la construcción del modelo se utilizó la implementación de la librería scikit-learn con los parámetros por defecto. Este algoritmo no requiere mayor configuración.
2. **Random forest**, es un tipo de algoritmo de ensamblaje y consiste en una combinación de árboles predictores [16]. La definición de los árboles se basa en un particionamiento binario recursivo del espacio de predicción en donde el nodo raíz comprende todo el espacio de predicción y cada árbol contempla un conjunto de variables aleatorias [17]. Breiman propone el uso de Random Forest no solo para clasificación sino también

para regresión [16]. En este caso el valor se obtiene mediante el promedio de los valores predichos por los árboles.

En la construcción se implementó una función que permite visualizar la evolución del error generado por el algoritmo en base a la profundidad del árbol. De esta manera el usuario final, puede identificar la mejor profundidad para el modelo.

La implementación realizada de este algoritmo contempla como valores: una profundidad de 5 niveles y un total de 50 árboles para las predicciones. Se configura también una semilla a través del parámetro `random_state` con el fin de que sus resultados sean reproducibles. Adicionalmente, se utiliza el error cuadrado medio “mse” (Mean Square Error) para estimar la calidad de las divisiones. El usuario puede ajustar estos valores de así considerarlo necesario.

3. **Máquinas de vector de soporte (regresión)**, de acuerdo a Pedregosa [18] las máquinas de vector de soporte son un conjunto de herramientas de aprendizaje automático que se utilizan tanto para clasificación como para regresión y detección de anomalías. Su principal ventaja es la efectividad que tiene en espacios de gran número de dimensiones. La implementación específica para regresiones se denomina Support Vector Regression. Este tipo de algoritmos tienen varias alternativas de funciones de kernel que permiten manejar tanto esquemas lineales como no lineales. Para el presente trabajo se seleccionó un kernel no lineal denominado Radial Basis Function (RBF) con lo cual se complementa el abanico de opciones brindadas al usuario. Adicionalmente, se utilizó el valor de 0.8 como parámetro de regularización, el cual utiliza el cuadrado de la penalización L2 [18]. Se usa el valor de epsilon por defecto que es de 0.1, el cual genera un margen en el que, no se penaliza la diferencia entre el valor predicho y el esperado [18]. Finalmente, para el parámetro gamma se consideró el uso de la opción “scale” la cual considera el número de características y la varianza que existe en los valores.

4. **Red neuronal artificial**, son modelos computacionales inspirados por la estructura de las redes neuronales del cerebro, en donde muchas neuronas, organizadas en capas, pueden estar interconectadas y transportan el resultado de sus cálculos a las siguientes neuronas de la red. Cada neurona toma las salidas de las neuronas anteriores (conectadas a esa neurona) y realiza las sumas de esas entradas ponderándolas por ciertos pesos. El resultado de la suma ponderada es utilizado como entrada para calcular la operación definida en la neurona como función de activación y el resultado se pasa a la siguiente neurona; a este proceso se denomina feedforward [15].

Para la construcción de la red se experimentó con varias redes neuronales y se estableció un modelo compuesto por una capa de entrada en donde el número de neuronas varía en función de las características de la demanda; para el caso de aplicación son 4. Después, vienen 2 capas Densas con activador ReLU e inicializador “glorot_uniform” y finalmente 1 capa de salida densa con una sola neurona para realizar la regresión, en este caso no se usa activador. En la figura 6. Se puede visualizar la red de manera gráfica.

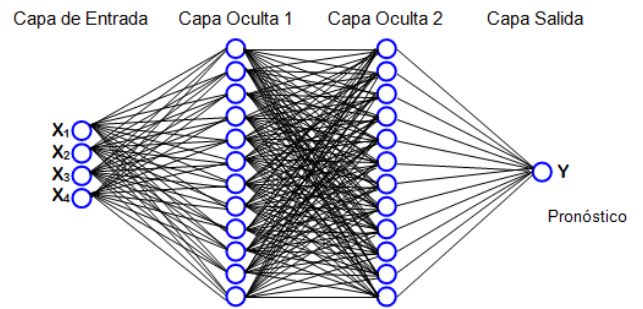


Fig. 6. Red neuronal implementada para pronósticos de capacidad tecnológica. En la imagen se visualiza el diseño de la red neuronal implementada como parte del sistema IACT para generar pronósticos de capacidad tecnológica. Fuente: Elaboración Propia

Para la función de optimización se contempló el algoritmo Adam con sus parámetros por defecto y como función de pérdida se utiliza el error cuadrático medio “mse”. Adicionalmente se usa el 20% de los datos de entrenamiento para la validación del modelo.

A más del modelo se implementó la generación de un gráfico que permite visualizar la comparación del error cuadrático medio (mse) de los datos de entrenamiento con el mse de los datos de validación, esto permite al usuario identificar los casos en que existe sobreajuste.

Implementación de interfaces de usuario:

En la solución planteada el usuario interactúa directamente con dos entornos, por un lado, accede a las pantallas de la aplicación web java del sistema IACT y por otro lado accede a los modelos python a través de Jupyter Notebook. En la Figura 7 se pueden apreciar las tecnologías utilizadas para la implementación de las interfaces de usuario:

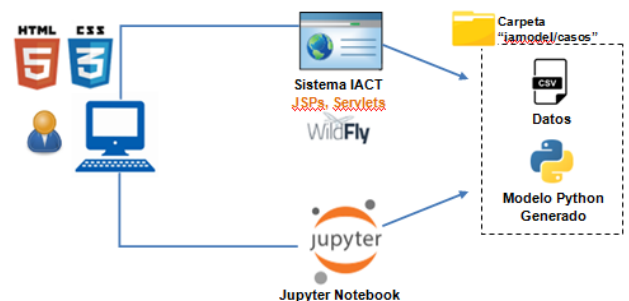


Fig. 7. Tecnologías utilizadas para la implementación de la interfaz de usuario. En la imagen se aprecian las tecnologías de los tres ámbitos principales: el cliente (navegador) el servidor WildFly y el servidor Jupyter. Fuente: Elaboración propia

Como se puede apreciar en la Figura 7, entre el servidor WildFly y el Notebook, se tiene una carpeta mediante la cual se comparten los archivos de datos y el código python generado por el sistema IACT, para que el usuario pueda utilizarlo desde Jupyter.

A modo de ejemplo se presenta en la Figura 8 la pantalla del sistema IACT correspondiente al paso final de generación del modelo. En la pantalla se puede apreciar el menú de opciones disponibles, el caso sobre el que se está trabajando y el mensaje de éxito en el cual se encuentra el link que permite abrir el modelo desde Jupyter Notebook.

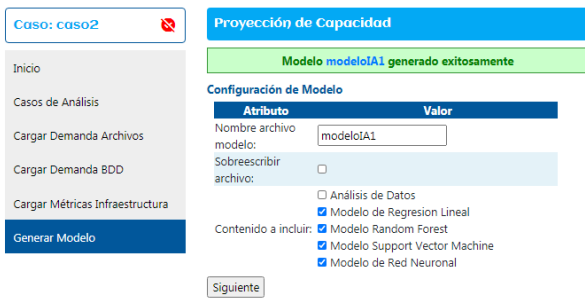


Fig. 8. Pantalla final de la generación del modelo inteligente. En la imagen se visualiza la estructura de las pantallas implementadas y el mensaje de éxito en donde se encuentra el link que permite abrir el modelo en Jupyter Notebook. Fuente: Capturas del sistema desarrollado

Adicionalmente, como se puede observar en la Figura 8, el usuario puede escoger los contenidos que desea incluir en el modelo generado.

V. EVALUACIÓN Y RESULTADOS

Para la evaluación del modelo se ha considerado principalmente la métrica del error cuadrático medio o mean square error (MSE) como se lo conoce en inglés. Este indicador se define como la suma del cuadrado de la diferencia entre la calificación prevista y la calificación real [19]. Esta función se utilizó tanto durante el entrenamiento como objetivo de minimización, como para la comparación de todos los modelos.

Adicionalmente, también se utilizó la raíz cuadrada del MSE, indicador denominado RMSE (Root Mean Square Error), por cuanto es un valor más interpretable ya que se encuentra en las mismas medidas que los datos originales [19].

Finalmente, se obtuvo la métrica del error absoluto medio o mean absolute error (MAE) con la cual en lugar de tomar el cuadrado se calcula el valor absoluto de la diferencia del valor real y del predicho. Estos valores también se encuentran en las mismas dimensiones de los datos.

A continuación, se presentan los siguientes resultados:

- Evaluación 1, explica la evolución del error MSE respecto a la profundidad de Random forester.
- Evaluación 2, indica la evolución del MSE de la red en cada época del entrenamiento.
- Evaluación 3, indica la comparación de los resultados de todos los cuatro algoritmos.

Evaluación 1, Random forest mse vs profundidad

En el caso de Random Forest, se analiza la evolución que existió del error cuadrático medio respecto del nivel de profundidad. Como se puede apreciar en la Figura 9, en lugar de disminuir el error se identifica que a mayor profundidad el error se incrementa. Por lo antes indicado, en este caso, se ajusta la profundidad del algoritmo a 2 niveles.

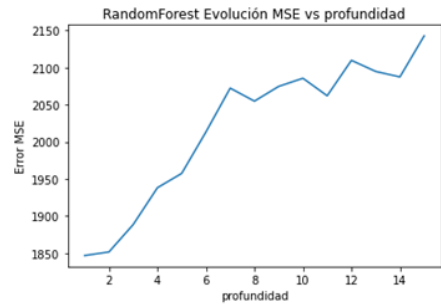


Fig. 9. Análisis del error cuadrático medio respecto a profundidad de Random forest. En la imagen se puede identificar que el error se incrementa cuando se tiene mayor profundidad en el algoritmo. Fuente: Grafico de capturado en Jupyter Notebook

Evaluación 2, red neuronal - evolución mse vs época

En el caso de la red neuronal, se analiza la evolución del error cuadrático medio con los datos de entrenamiento y de validación a fin de identificar si existe un posible sobre ajuste en el modelo.

Como puede visualizarse en la Figura 10, el resultado obtenido indica que el error se reduce por cada época y no hay sobre ajuste.

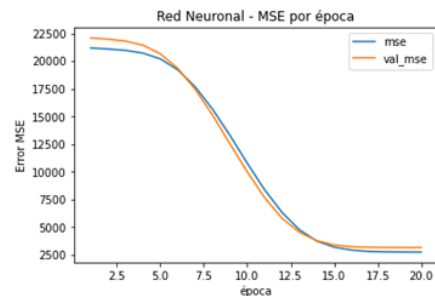


Fig. 10 Análisis del error cuadrático medio por época en entrenamiento de la red neuronal. Como se puede apreciar la red neuronal tiene un comportamiento adecuado ya que por cada época se reduce el error cuadrático medio y el comportamiento es similar en los datos de entrenamiento y de validación. Fuente: Grafico de capturado en Jupyter Notebook

Evaluación 3, comparación de modelos

Para concluir, la evaluación en la tabla 1 resume el resultado de los indicadores obtenidos con los cuatro algoritmos.

Tabla 1. Resultado comparativo de modelos predictivos para la métrica "MemoriaUsada/JVM"

Algoritmo \ Métrica	Regresión Lineal	Random Forest	SVR	Red Neuronal
MSE	1888,39	1851,72	1976,69	2371,68
RMSE	43,46	43,03	44,46	48,69
MAE	36,60	36,17	35,85	39,54

Fuente: Elaboración propia, a partir de valores presentados en el notebook de los modelos

En base a los resultados obtenidos en el error cuadrático medio, se puede identificar que el menor valor se encuentra para el algoritmo Random Forest; sin embargo, los otros algoritmos también presentan resultados cercanos. De los cuatro algoritmos el que tiene un peor resultado es la red neuronal.

Al analizar los errores en base al Error Absoluto Medio (MAE por sus siglas en inglés), quien presenta mejores resultados son las Máquinas de Vector de Soporte (Regresión) seguido por

Random Forest. Considerando que el MSE penaliza en mayor medida los errores se opta para este caso por el estimador Random Forest.

El valor máximo que puede tener la métrica es de 247,64. Al comparar la raíz del error cuadrático medio que es de 43,03 con el valor máximo se identifica que el error representa el 17.37% de este valor.

VI. DISCUSIÓN

En base a los resultados obtenidos se confirma que se cumplió el objetivo establecido en el presente trabajo.

Es importante mencionar que, por cada métrica, se deben ajustar los parámetros de los modelos a fin de asegurar que estos respondan al caso previsto.

VII. CONCLUSIONES

Es un factor de vital importancia, que las organizaciones puedan identificar de manera fácil y oportuna la capacidad de la infraestructura tecnológica necesaria para soportar su operación actual y futura. Este análisis conlleva gran complejidad y dificultad ya que requiere de conocimientos profundos de tecnología y del negocio.

El trabajo realizado permitió construir una herramienta de software que haciendo uso de técnicas de aprendizaje automático, facilita el análisis de la capacidad de la infraestructura tecnológica.

Para abordar el problema se exploró las mejores prácticas de gestión de la capacidad y se identificó la fuerte relación que existe con la gestión de la demanda. También se realizó una búsqueda exhaustiva de investigaciones relacionadas a la aplicación de la inteligencia artificial a la gestión de capacidad tecnológica y se pudo notar que es un campo poco explotado. Los trabajos identificados tienen una orientación diferente, mientras algunos contemplan la administración de nubes públicas y otros tratan de infraestructura de telefonía o eléctrica, ninguno aborda la identificación de capacidad de infraestructura tecnológica considerando su relación con la demanda transaccional.

Para el desarrollo se utilizó la metodología CRISP-DM, la cual parte del entendimiento del negocio, continúa con el entendimiento y preparación de los datos y con esta información se desarrolla el modelamiento, se lo evalúa y finalmente se implementa las facilidades para que el usuario final pueda hacer uso de los modelos creados.

Para el entendimiento de negocio, se definió un caso concreto de aplicación denominado Administración de Libros. Para el cual se construyó un sistema web y un cliente que simula la carga transaccional. Con el caso definido, se levantaron y establecieron los siguientes requerimientos de la solución: gestión de casos de análisis, carga de datos de demanda, carga de métricas de capacidad y la generación y uso del modelo inteligente. Después, se plantearon e implementaron los esquemas de monitoreo de la infraestructura y al correr la simulación de carga, se tomaron las métricas de uso de recursos como: memoria, procesamiento y almacenamiento.

A continuación, se analizó la información recolectada y se implementó la preparación de los datos en dos fases, la primera para estandarizar la información a formato de archivos CSV (valores separados por comas) y la segunda que consolidó y preparó los datos para los modelos inteligentes.

Con los datos listos, se seleccionaron e implementaron los siguientes modelos de aprendizaje supervisado: Regresión Lineal, Random Forest, Máquinas de Vector de Soporte (Regresión) y Red Neuronal Artificial. Después se corrió los modelos para la medida de uso de memoria JVM del caso de administración de libros, y se analizó los resultados utilizando tres métricas de calidad: error cuadrático medio, raíz del error cuadrático medio y error absoluto medio. Como resultado se obtuvo que el modelo que presentaba el menor error cuadrático medio fue Random Forest (1851,72), seguido por la Regresión Lineal (1888,39) y las máquinas de vector de soporte con regresión (1976,69), el mayor error se tuvo en la Red Neuronal (2371,68). Por otro lado, el algoritmo que presentó el menor error absoluto medio fue máquinas de vector de soporte con regresión (35,85), seguido por Random Forest (36,17) y Regresión lineal (36,6). De igual manera el de mayor error fue la red neuronal con (39,54).

Al comparar la raíz del error cuadrático medio de Random Forest (43,03 MB), con el mayor valor de la métrica (247,64 MB) se obtuvo un porcentaje de 17.37%, cumpliendo así, el objetivo general de la investigación que era, crear una herramienta inteligente que permita proyectar los requerimientos de capacidad con un error menor al 20% del valor máximo de la métrica.

El sistema implementado IACT (Inteligencia Artificial para Capacidad Tecnológica) fue probado por tres usuarios expertos y cuenta con interfaces de usuario robustas y con diseños responsivos e intuitivos. Durante las pruebas, se confirmó que el sistema facilita la carga y la preparación de los datos de demanda y de métricas de uso de recursos de infraestructura. Adicionalmente, IACT es capaz de generar un modelo completo en formato Notebook que incluye: el análisis, la preparación de datos, los modelos y la evaluación de resultados. Dicho modelo es accesible al usuario para su afinamiento y uso mediante Jupyter Notebook.

Una vez que ya se confirmó la validez de la solución, en base a la información del caso de aplicación, se plantea que esta herramienta sea instalada y utilizada para analizar datos en un entorno real, el cual podría incluir métricas de componentes de infraestructura adicionales. En base a la retroalimentación que se obtenga del uso del sistema en el entorno real, se podría ajustar los modelos propuestos o plantear la generación de modelos adicionales que permitan expandir y mejorar las capacidades predictivas de la solución.

El esquema de plantillas y de generación de modelos predictivos que se implementó en el aplicativo web java, puede servir de referencia para otras iniciativas de inteligencia artificial, en las cuales se requiera integrar ambientes inteligentes de análisis de información construidos en python con aplicaciones web desarrolladas en otros lenguajes.

El sistema implementado para el caso de aplicación puede ser utilizado para generar otros conjuntos de datos que permitan realizar mayor investigación de Inteligencia Artificial aplicada a la gestión de capacidad tecnológica.

REFERENCIAS

- [1] Expansión. (2019, octubre 25). La receta para maximizar la inversión en tecnología. Expansión.com. <https://www.expansion.com/economia-digital/companias/2019/10/25/5db1df50468aeba9468b46fb.html>.
- [2] Naciones Unidas. (2020). E-Government Survey 2020 Digital Government in the Decade of Action for Sustainable Development. [publicadministration.un.org/egovkb/Portals/egovkb/Documents/un/2020-Survey/2020%20UN%20E-Government%20Survey%20\(Full%20Report\).pdf](http://publicadministration.un.org/egovkb/Portals/egovkb/Documents/un/2020-Survey/2020%20UN%20E-Government%20Survey%20(Full%20Report).pdf)

- [3] Office of Government Commerce. (2007a). *ITIL Service Design*. The Stationery Office.
- [4] Félix-Sánchez, A., & Calvo-Manzano, J. A. (2015). Comparison of models and standards for implementing IT service capacity management. *Revista Facultad de Ingeniería Universidad de Antioquia*, 74, 86-95.
- [5] Office of Government Commerce. (2007b). *ITIL Service Strategy*. The Stationery Office.
- [6] Jiang, Y., Perng, C.-S., Tao, L., & Rong, C. (2012, abril). Intelligent Cloud Capacity Management. 2012 IEEE Network Operations and Management Symposium, Maui, HI, USA. <https://doi.org/10.1109/NOMS.2012.6211941>
- [7] Jiang, Y., Perng, C.-S., Tao, L., & Rong, C. (2013). Cloud Analytics for Capacity Planning and Instant VM Provisioning. *IEEE Transactions on Network and Service Management*, 10(3), 312-325. <https://doi.org/10.1109/TNSM.2013.051913.120278>
- [8] Pérez-Romero, J., Sallent, O., Ferrús, R., & Agustí, R. (2015). Artificial Intelligence-based 5G network capacity planning and operation. 2015 International Symposium on Wireless Communication Systems (ISWCS), 246-250. <https://doi.org/10.1109/ISWCS.2015.7454338>
- [9] Seyedeh, N. F., Deo, R. C., Shojafar, M., Conti, M., & Shamshirband, S. (2018). Computational Intelligence Approaches for Energy Load Forecasting in Smart Energy Management Grids: State of the Art, Future Challenges, and Research Directions - ProQuest. *Journal Energies*, 11(3). <https://doi.org/DOI:10.3390/en11030596>
- [10] Huang, J., Tang, Y., & Chen, S. (2018). Energy Demand Forecasting: Combining Cointegration Analysis and Artificial Intelligence Algorithm - ProQuest. *Mathematical Problems in Engineering*, 2018. <https://doi.org/DOI:10.1155/2018/5194810>
- [11] Eremia, M., Liu, C.-C., & Edris, A.-A. (2016). *Advanced Solutions in Power Systems: HVDC, FACTS, and Artificial Intelligence*. John Wiley & Sons.
- [12] Veit, A., Xu, Y., Zheng, R., Chakraborty, N., & Sycara, K. (2014). Demand Side Energy Management via Multiagent Coordination in Consumer Cooperatives. *Journal of Artificial Intelligence Research*, 50, 885-922. <https://doi.org/10.1613/jair.4416>
- [13] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0 Step-by-step data mining guide*.
- [14] Project Gutenberg. (2020, agosto 14). *Free eBooks—Project Gutenberg*. <https://www.gutenberg.org/>
- [15] Shai, S.-S., & Shai, B.-D. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.
- [16] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1).
- [17] Cutler, A., Cuttler, D., & Stevens, J. (2011). *Random Forests*. https://doi.org/10.1007/978-1-4419-9326-7_5
- [18] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [19] Dua, R., Ghotra, M. S., & Pentreath, N. (2017). *Machine Learning with Spark—Second Edition*. Packt Publishing.