

Proyecto de título Ingeniería Civil Informática - Ingeniería Civil Industrial

*"Diseño y desarrollo de plataforma de generación datos para la mejora del
proceso de ventas de AWS Supply Chain"*

Alumno: Cristóbal Drago Pérez

Supervisor: Felipe Chirinos

24 de diciembre, 2023

Resumen ejecutivo

Amazon Web Services (AWS) es una plataforma de servicios en la nube ofrecida por Amazon. El foco principal de AWS es brindar a terceros la posibilidad de externalizar infraestructura de tecnología y poder así acceder a una gran red de soluciones, herramientas y servicios, habilitándolos así a construir y gestionar aplicaciones, almacenar datos y utilizar servicios especializados sin la necesidad de invertir en infraestructura y hacerlo de manera eficiente, segura y escalable.

Dentro de AWS están los equipos de Solutions Architects (SAs), encargados de asesorar a clientes velando por las buenas prácticas de arquitectura disponibles en el Well-architected Framework de AWS y de verificar la viabilidad técnica de las soluciones propuestas por AWS mediante demostraciones. Lo anterior ha sido un desafío en el servicio AWS Supply Chain: el proceso de demostraciones le toma un 20% del tiempo que dispone un SA para estas tareas y no tienen educación suficiente para desarrollar una demostración confiable.

Desde este foco, se propuso como objetivo mejorar el proceso de venta de AWS Supply Chain mediante la disminución en un 50% del tiempo invertido por un SA en las demostraciones de este servicio. Para lograr este objetivo, se elaboraron medidas de desempeño que fueron utilizadas para guiarnos y analizar los resultados del proyecto.

Se seleccionó 1 de 3 soluciones propuestas con el objetivo de cumplir con los objetivos específicos del proyecto. Esta solución consiste en el desarrollo de una plataforma de generación de datos soportados por AWS Supply Chain, datos que siguen un esquema de una cadena de suministro predefinida y son totalmente personalizables para el caso específico del cliente.

Abstract

Amazon Web Services (AWS) is a cloud services platform offered by Amazon.com. The main focus of AWS is to provide third parties the ability to outsource technology infrastructure and thus be able to access a large network of solutions, tools and services, enabling organizations and developers to build and manage applications, store data and use specialized services without the need to invest in hardware and to do so in an efficient, secure and scalable way.

Within AWS are the Solutions Architects (SAs) teams, in charge of advising customers on the best architecture practices available in the AWS Well-architected Framework and increasing the visibility of services through demonstrations to increase sales. This has been an arduous challenge in the AWS Supply Chain project: the demo process takes 20% of the time an SA has available for these tasks and they are not sufficiently educated about the supply chain to generate reliable demos for customers. Thus, there is significant pain in one of the essential links in the AWS sales process for the Supply Chain service.

From this focus, the objective was set to improve the AWS Supply Chain sales process by reducing by 50% the time invested by an SA in the demonstrations of this service in a period of 3 months. To achieve this objective, different performance measures were developed that were used to guide us in the process and analyze the results of the project.

We selected one solution over 3 proposed solutions in order to mitigate the pains of the company and meet the specific objectives of the project. This solution is the development of a data generation platform supported by AWS Supply Chain, data that follows a predefined supply chain schema and is fully customizable for the customer's use case: country where it is located and industry it covers.

Índice

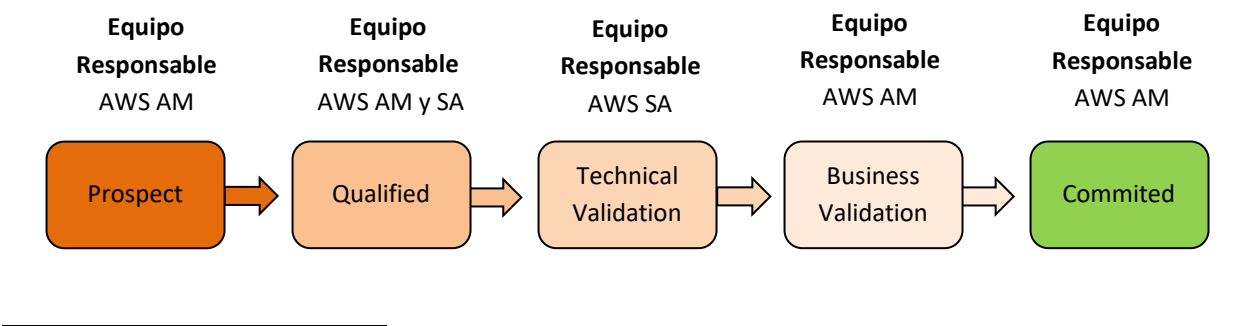
Introducción	5
Identificación de la oportunidad	7
Objetivos	9
Objetivo SMART	9
Objetivos específicos	9
Estado del arte	10
Propuestas de solución, solución escogida y riesgos	13
Propuestas de solución y solución escogida	13
Matriz de riesgos	15
Metodologías	17
Planificación	18
Evaluación Económica	19
Medidas de desempeño	21
Desarrollo del proyecto	23
Diseño y desarrollo de modelo de entidad relación para AWS Supply Chain	23
Diseño y desarrollo front-end	25
Desarrollo back-end	26
Desarrollo algoritmos de generación de datos con AWS Lambda	27
Generación de datos base con Amazon Bedrock	28
Integración con AWS Step Function y Amazon API Gateway	29
Implementación	30
Resultados cualitativos y cuantitativos	32
Medida de desempeño 1	32
Medida de desempeño 2	32
Medida de desempeño 3	33
Medida de desempeño 4	33
Medida de desempeño 5	33
Conclusión	34
Anexos	35
Referencias	44

Introducción

Amazon Web Services (AWS) es una plataforma de servicios de software en la nube ofrecida por Amazon, siendo uno de los líderes en lo que conocemos como “cloud computing” al tener un 32% de participación en el mercado según datos del segundo cuartil del 2023¹. El foco principal de AWS es brindar a terceros la posibilidad de externalizar la infraestructura de tecnología necesaria para desarrollar aplicaciones de diferentes tipos, proporcionando una alta gama de servicios, herramientas y soluciones que permiten a organizaciones y desarrolladores construir y gestionar aplicaciones, almacenar datos y ejecutar servicios de manera eficiente, segura y escalable.

El proceso de venta de servicios de AWS se puede resumir en 5 eslabones principales (revisar Imagen 1):

1. “Prospect”: AWS identifica una oportunidad con un cliente. Esta puede estar activa (viene desde el cliente) o latente (identificada por un AWS Account Manager (AM)). En cualquiera de los dos casos, el representante de AWS será el AM.
2. “Qualified”: Account Team (AM y AWS Solutions Architect (SA)) se ha comprometido con el cliente para discutir la viabilidad y definir los requisitos de la oportunidad. El cliente ha aceptado que la oportunidad es real, de interés y puede resolver necesidades comerciales/técnicas clave.
3. “Technical Validation”: la solución es validada técnicamente por el cliente a través de una combinación de demostraciones y sesiones de diseño arquitectónico por parte del SA.
4. “Business Validation”: las partes interesadas del negocio han comunicado un acuerdo sobre la viabilidad financiera de la solución a través de una serie de discusiones sobre modelos de costos y análisis de ROI/TCO (“Return Over Investment” y “Total Cost of Ownership”) con el AM.
5. “Committed”: cliente ha aceptado la solución y se compromete a avanzar en términos de tecnología, arquitectura y economía.



¹ Richter, F. (2023, August 8). *Infographic: Amazon Maintains Lead in the Cloud Market*. Statista. Retrieved October 18, 2023, from <https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/>

Imagen 1: proceso de venta AWS resumido en 5 eslabones

El alumno estuvo inmerso en el área de los Solutions Architects (SAs). Ellos, son los encargados de asesorar a los clientes de AWS en sus arquitecturas en la nube, siempre velando por las buenas prácticas de AWS referidas en el Well-Architected Framework². Además, son también los responsables de validar las oportunidades de venta (“Technical Validation”) mediante demostraciones con potenciales clientes. En estas sesiones, se explica y demuestra un servicio de AWS mediante la práctica, esperando que se enamoren del servicio visualizando los beneficios de este. La forma en la que un SA debe distribuir su tiempo sigue una regla de 50-35-15:

1. 50% de su tiempo en “Customer Engagement”: actividades enfocadas en su Account Team, clientes (un SA tiene múltiples clientes), y empresas partner de AWS.
2. 35% de su tiempo en “Sharpening the Saw”: desarrollo de conocimiento del SA.
3. 15% de su tiempo en “General Activities”: tareas administrativas, organización interna, reuniones de equipo, etc.

Dentro de los servicios brindados por AWS está AWS Supply Chain, proyecto aún en desarrollo donde el alumno desarrolló su proyecto de práctica. El objetivo principal del servicio es brindar a terceros la posibilidad de usar la inteligencia logística de Amazon.com en sus negocios, “mitigando así los riesgos de la cadena y optimizando sus costos utilizando aprendizaje automático”³. Es fundamental mencionar que AWS Supply Chain no es una herramienta de gestión o control de la cadena de suministro que se preocupa de manipular y almacenar los datos de ella, más bien es una herramienta que debe ser alimentada con datos externos de una cadena de suministro para así visualizar los desafíos presentes en nuestra cadena y entregar recomendaciones para mejorar el rendimiento de ella. Dentro de las características del servicio están: recomendaciones (“Insights”) y planificación de demanda.

En base a una encuesta realizada de manera interna a SAs de AWS (revisar punto 1 del anexo), que tuvo como objetivo entender la postura y desafíos respecto al servicio en cuestión, se determinó que existe una oportunidad para AWS de mejorar el proceso de venta del servicio de AWS Supply Chain debido al

² AWS - Well-Architected Framework. (2015, October 1). *AWS Well-Architected - Build secure, efficient cloud applications*. Amazon AWS. Retrieved October 10, 2023, from <https://aws.amazon.com/architecture/well-architected/>

³ AWS Supply Chain. (2022, November 29). *Supply Chain Optimization — AWS Supply Chain — Amazon Web Services*. Amazon AWS. Retrieved October 07, 2023, from <https://aws.amazon.com/aws-supply-chain/>

excesivo tiempo que requiere para los SAs la preparación y realización de demostraciones, además de la inexistencia de casos de uso claros, realistas, y específicos para poder demostrar los beneficios que tiene para un cliente.

Identificación de la oportunidad

La oportunidad de mejora en la empresa nace de la brecha de material que disponen los SAs para poder utilizar en las demostraciones del servicio AWS Supply Chain para oportunidades en “Technical Validation” del proceso de venta, limitando así la posibilidad de mostrar el funcionamiento completo y los beneficios del servicio según el caso del cliente. De esta manera, el proceso de venta se ve ralentizado y no genera confianza con el cliente: el SA debe desarrollar demostraciones que simulen una cadena de suministro sin un fundamento sólido que represente los desafíos reales de un cliente. Lo mencionado, fue conclusión de una encuesta que tuvo como objetivo analizar y descubrir los dolores y brechas de AWS. Los resultados fueron los siguientes.

En primer lugar, solamente el 34% de las oportunidades de venta del servicio termina siendo efectiva (se consideró una venta efectiva como el paso a “Commit” de un cliente en el proceso de venta).

En segundo lugar, el 76% de los encuestados señala que les toma 14 horas desarrollar material técnico para una demostración de este servicio, con el objetivo de mostrar sus funcionalidades y beneficios. Esta tarea es clasificada como “Customer Engagement” en la distribución de tiempo de un SA, englobada dentro del 50% o 72 HH (Horas Hombre) que él dedica a este tipo de tareas. Así, la cantidad de tiempo que dedica un SA a la elaboración de contenido técnico para un posible cliente de este servicio es extremadamente alta (considerando que un SA tiene múltiples clientes), representando un 20% del total de las horas mensuales disponible para este tipo de tareas.

En tercer lugar, dentro de los desafíos con los que se encuentran los SAs al momento de realizar una demostración, los que más destacaron fueron la complejidad de la integración de datos hacia la plataforma (67% de los encuestados), la complejidad de los requerimientos del esquema de cadena de suministro requerido por AWS Supply Chain (74% de los encuestados), y la complejidad de generación de datos para su uso (56% de los encuestados). Por otro lado, solamente el 7% señaló que su área de conocimiento está relacionada con la cadena de suministro, demostrando así una falta de educación de los SAs respecto a este tema.

Por último, existe gran variabilidad entre los países e industrias de los clientes que han mostrado interés por el servicio. Se obtuvo un total de 12 países y 10 industrias, observando así que el interés por el servicio AWS Supply Chain es contingente a diferentes industrias y países.

Objetivos

Objetivo SMART

El objetivo general SMART del proyecto es el siguiente (el desglose de las siglas “SMART” disponible en Anexo 11):

“Disminuir un 50% el tiempo invertido por un SA en las oportunidades en “Technical Validation” del servicio AWS Supply Chain mediante el rediseño del proceso de demostraciones en un periodo de 3 meses”

Objetivos específicos

Para lograr el objetivo SMART, se deberá cumplir con los siguientes objetivos específicos:

1. Rediseñar el proceso de desarrollo de demostraciones del servicio para demorar un total de 6 horas o menos, enfocando el proceso en ser confiable, automatizado y reutilizable.
2. Diseñar demostraciones que cumplan con los requisitos de AWS Supply Chain para las características de planificación de demanda y recomendaciones.
3. Abarcar a los 4 países más contingentes para AWS a nivel latinoamericano (países donde AWS tiene sede): Argentina, Chile, Colombia, y Brasil.

Estado del arte

Para la evaluación de soluciones en el proyecto, se analizó el estado del servicio, casos en la industria, e investigaciones en las que se han abordado problemáticas similares a las mencionadas. Así, se evaluó el estado del arte en cuatro temáticas: generación de datos sintéticos de cadena de suministro para AWS Supply Chain, requerimientos de AWS Supply Chain, distribuciones de probabilidad representativas de demanda, herramientas de generación de datos y datos públicos de cadena de suministro.

En primer lugar, existe una única forma de generar una base de datos poblada con datos sintéticos que puede ser utilizada para la demostración del servicio AWS Supply Chain: consta de un Jupyter Notebook que genera los datos transaccionales necesarios para demostrar el servicio guardándolos en un archivo “csv” por cada entidad ⁴. Fue desarrollado de manera interna para el lanzamiento del servicio el año 2022 y consta de las entidades necesarias para demostrar solamente la característica de planificación de demanda y recomendaciones. Estos datos fueron desarrollados para 8 productos falsos de una empresa con localidades en Estados Unidos (USA). Para generar la demanda de cada producto y su respectivo pedido de inventario se utilizó una distribución normal con los mismos parámetros para los 8 productos (media y desviación estándar). El problema con estos datos son los siguientes: es aplicable solamente a USA, no es contingente a una industria específica (los productos son falsos y se utilizan los mismos parámetros para la generación de demanda), y la generación sintética de demanda irreal.

En segundo lugar, como se planteó en la sección de identificación de la oportunidad, uno de los grandes dolores identificados fue la complejidad de la integración de datos y la complejidad del desarrollo del esquema de cadena de suministros para poder poblar con datos. Respecto a lo primero, actualmente AWS Supply Chain soporta solo las siguientes integraciones a fuentes de datos: Amazon S3, “Electronic Data Interchange” (EDI), SAP S/4HANA, y SAP ECC 6.0 ⁵. De ellas, la única disponible para utilización por los SAs es Amazon S3: EDI no aplica al caso y SAP requiere licencias pagadas. Respecto a la complejidad del esquema de cadena de suministro, actualmente, este debe seguir minuciosamente las especificaciones

⁴ AWS: Confidencial. (2022, Diciembre 10). *AWS Supply Chain - Dataset*.

⁵ AWS. (2022, Noviembre 29). *Connecting to a data source - AWS Supply Chain*. AWS Documentation. Retrieved Octubre 01, 2023, from <https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/data-connections.html>

de nombres de las entidades y atributos definidas por el servicio en su documentación ⁶. De no ser así, AWS Supply Chain es incapaz de operar de manera correcta.

En tercer lugar, como se menciona en el libro “Inventory Control and Management” escrito por Donald Waters, existen modelos de control de inventario de demanda independiente conocida y modelos que asumen demanda variable siguiendo una distribución de probabilidad conocida⁷. En el libro en cuestión, se analizan modelos en los que tanto la demanda como el “lead time” distribuyen según una distribución normal de probabilidad, siendo así ambos valores inciertos. El modelo utilizado para el caso descrito nos permite calcular la desviación estándar del “lead time demand” (demanda de tiempo de entrega) en base a los parámetros (media y desviación estándar) de la demanda y el “lead time”, habilitándonos así a calcular el inventario de seguridad, tamaño del pedido, y “reorder level” (nivel de inventario para volver a pedir) ⁸.

En cuarto lugar, la Inteligencia Artificial Generativa (GenAI) ha causado un cambio abrupto en la disponibilidad de información y la interacción con ella por parte del público, especialmente con el lanzamiento de ChatGPT 3 y 3.5 de manera pública por parte de OpenAI el año 2022 ⁹. Sin embargo, como se menciona en el artículo “The Promises and Perils of Generative AI in Education” ¹⁰, existen ciertos riesgos respecto al uso de esta herramienta, entre ellos el sesgo hacia algunos temas y la entrega de información inexacta. Además, es fundamental mencionar que muchas compañías prohíben estrictamente la utilización de ChatGPT, tal es el caso de AWS. No solamente OpenAI está en la vanguardia de este tema, AWS con su servicio “Amazon Bedrock” también busca otorgar a sus clientes las herramientas necesarias para que generen sus propias herramientas de IA, mitigando las limitaciones de este y elaborando herramientas privadas al cliente poniendo a su disposición múltiples modelos de “Large Language Models” (LLMs) ¹¹. Por otro lado, un caso de uso de GenAI es la generación de datos sintéticos,

⁶ AWS. (2022, Noviembre 29). *Data entities supported in AWS Supply Chain - AWS Supply Chain*. AWS Documentation. Retrieved Octubre 01, 2023, from <https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/data-model.html>

⁷ Waters, C. D. J. (2003). *Inventory Control and Management* (2nd ed.), p. 147-148. Wiley.

⁸ Waters, C. D. J. (2003). *Inventory Control and Management* (2nd ed.), p. 180-181. Wiley.

⁹ OpenAI. (2022, Noviembre 30). *Introducing ChatGPT*. OpenAI. Retrieved Noviembre 27, 2023, from <https://openai.com/blog/chatgpt>

¹⁰ Teach For America. (2023, Agosto 11). *The Promises and Perils of Generative AI in Education: TFA's Evolving Perspective*. Teach For America. Retrieved Diciembre 02, 2023, from <https://www.teachforamerica.org/one-day/ideas-and-solutions/the-promises-and-perils-of-generative-ai-in-education-tfas-evolving>

¹¹ AWS - Amazon Bedrock. (2023, Septiembre 28). *Foundational Models - Amazon Bedrock - AWS*. Amazon AWS. Retrieved Noviembre 27, 2023, from <https://aws.amazon.com/bedrock/>

tal y como se desarrolla en artículo “Generative Adversarial Networks and Synthetic Patient Data”, donde se analiza el uso de GenAI para crear datos sintéticos de pacientes y así priorizar el anonimato de estos. En este, se concluye que el desafío principal de su utilización para elaborar datos sintéticos está en garantizar la representatividad estadística de ellos¹². De la mano de lo anterior, en el artículo “Role of Generative AI to Generate Synthetic Data” se exponen algunos de los beneficios y casos de usos de la generación de datos con GenAI, resaltando: datos sintéticos para entrenar modelos de Machine Learning (ML) cuando el dataset original es limitado, predicción de demanda de productos específicos en Supply Chain, y datos sintéticos para validar softwares y algoritmos¹³.

Por último, la disponibilidad de datos de cadenas de suministros es escasa. En el artículo “DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS” de Mendeley Data, se expone una base de datos completa de cadena de suministro que contiene datos de producción, niveles de stock, distribución, ventas, y aprovisionamiento¹⁴. Esta base de datos ha sido utilizada 2 veces para artículos académicos. Además, en el sitio web “Kaggle” se dispone de múltiples bases de datos con información respecto a niveles de stock, productos por órdenes, localidades, retraso de órdenes, etc. Entre ellos “Supply Chain Analysis”¹⁵ de Harsh Singh, “Comprehensive Supply Chain Analysis”¹⁶ de Dorothy Joel y “Supply Chain Data”¹⁷ de Laurin Bretcher.

¹² Xenonstack. (2023, October 2). *Generative Adversarial Networks and Synthetic Patient Data*. XenonStack. Retrieved December 1, 2023, from <https://www.xenonstack.com/blog/gans-and-synthetic-patient-data>

¹³ Kaur, J. (2023, Noviembre 21). *Top Generative AI Use Cases and Applications*. XenonStack. Retrieved Octubre 12, 2023, from <https://www.xenonstack.com/blog/generative-ai-use-cases>

¹⁴ Constante, Fabian; Silva, Fernando; Pereira, António (2019), “DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS”, Mendeley Data, V5, doi: 10.17632/8gx2fvg2k6.5

¹⁵ Singh, H. (2023, Enero). *Supply Chain Analysis*. Kaggle. Retrieved Diciembre, 2023, from <https://www.kaggle.com/datasets/harshsingh2209/supply-chain-analysis>

¹⁶ Joel, D. (2023, October). *Comprehensive Supply Chain Analysis*. Kaggle. Retrieved Diciembre, 2023, from <https://www.kaggle.com/datasets/dorothyjoel/us-regional-sales>

¹⁷ Bretcher, L. (2022, Julio). *Supply Chain Data*. Kaggle. Retrieved Diciembre, 2023, from <https://www.kaggle.com/datasets/laurinbrechter/supply-chain-data>

Propuestas de solución, solución escogida y riesgos

Propuestas de solución y solución escogida

A partir de los 3 objetivos específicos nace un subproblema que debe englobar al resto: el rediseño del proceso de desarrollo de demostraciones. Tomando en cuenta el análisis del estado del arte, se decidió armar tres posibles soluciones. Los resultados se encuentran en la gráfica de la Tabla 1.

Subproblema	Solución A	Solución B	Solución C
Rediseño del proceso de desarrollo de demostraciones	1. Definir parámetros realistas a partir datos públicos de cadenas de suministros para utilizar en la solución actual mencionada. Editar el Jupyter Notebook desarrolldo de manera interna para generar datos por cada país contingente.	2. Desarrollo de plataforma para generación automática de bases de datos relacionales que sigan el diseño requerido por AWS Supply Chain. Utilizar Amazon Bedrock para generar los datos y parámetros necesarios y Amazon S3 para la integración.	3. Utilizar base de datos públicas de una cadena de suministro y traducirlo al diseño requerido por AWS Supply Chain. Guardar los datos en Amazon S3 para facilitar la integración hacia el servicio.

Tabla 1: recopilación de problemas con sus respectivas soluciones

Con las soluciones ya definidas, se utilizó un método implementando la escala Likert (revisar Anexo 4) y se evaluaron 5 criterios según ella. A cada uno de los criterios se les asignaron ponderaciones diferentes para poder medir el impacto de cada una de las soluciones. Los criterios junto con su descripción se encuentran en la gráfica de la Tabla 2.

Criterio	Descripción	Ponderación
Nivel de complejidad para su uso	Para usar la solución cuanto conocimiento necesito para su implementación y uso	5%
Nivel de confianza	Cuán confiables o representativos de la industria son los datos ingestados a AWS Supply Chain.	15%
Nivel de automatización	Entrega la solución una automatización de una tarea que requiere arduo trabajo en la actualidad.	20%

Tiempo dedicado SAs	Tiempo que ahorrará la solución al SA una vez implementada	30%
Nivel de obsesión por el cliente	Cuán personalizable (país e industria) es la solución según el cliente (principio “Customer Obsession” de Amazon)	30%

Tabla 2: Criterios y ponderaciones para evaluar las diferentes soluciones propuestas

Posteriormente, se decidió que un equipo de tres SAs en Chile tuviese la tarea de evaluar las soluciones y así decidir cuáles de ellas llevar a cabo. Se obtuvieron los resultados de la Tabla 3 (revisar Anexo 4 para más detalles).

Solución	Resultados
1. Definir parámetros realistas a partir datos públicos de cadenas de suministros para utilizar en la solución actual mencionada. Editar el Jupyter Notebook desarrollado de manera interna para generar datos por cada país contingente.	2,3
<u>2. Desarrollo de plataforma para generación automática de bases de datos relacionales que sigan el diseño requerido por AWS Supply Chain. Utilizar Amazon Bedrock para generar los datos y parámetros necesarios y Amazon S3 para la integración.</u>	<u>4,2</u>
3. Utilizar base de datos públicas de una cadena de suministro y traducirlo al diseño requerido por AWS Supply Chain. Guardar los datos en Amazon S3 para facilitar la integración hacia el servicio.	3,6

Tabla 3: soluciones escogidas (subrayadas y en negrita)

Así, se decidió llevar a cabo la solución número 2.

Matriz de riesgos

Con el objetivo de llevar a cabo de la mejor manera el desarrollo e implementación de las soluciones, se generó una matriz para cuantificar el impacto de cada riesgo respecto a su probabilidad de ocurrencia en 4 criterios: nula o poco probable, medianamente probable, probable y altamente probable (revisar gráfica de la Tabla 4). De la misma manera, se elaboró una escala de nivel de preocupación para evaluar la acción a llevar a cabo en caso de que el impacto sea de un rango específico (revisar gráfica de la Tabla 5).

Probabilidad de ocurrencia	Impacto			
	Leve	Moderado	Grave	Crítico
Nula o poco probable	1	2	3	4
Medianamente probable	2	4	6	8
Probable	3	6	9	12
Altamente probable	4	8	12	16

Tabla 4: cuantificación del riesgo según impacto y probabilidad de ocurrencia

Preocupación	Rango	Acción
Baja	1 a 4	Crear plan de acción para mitigar riesgo
Media	5 a 9	Crear plan de acción para mitigar riesgo
Preocupante	10 a 16	Reevaluar viabilidad de la solución y elaborar dos nuevas soluciones

Tabla 5: métrica para decidir qué acción tomar en base a la cuantificación del riesgo

Todo lo anterior se realizó con el objetivo de definir mecanismos de mitigación de riesgos, tomando en cuenta la probabilidad de ocurrencia y el impacto de ellos. De esta manera, se obtuvieron los resultados de la Tabla 6.

Riesgo	Probabilidad	Impacto	Puntuación	Acción	Mitigación
Generación datos poco representativa	Probable	Grave	9	Crear plan de acción	Reevaluación algoritmo de generación de datos sintéticos. Asesorarse de la teoría y AWS Industry Advisor (especialista)
Error en ingesta de datos a AWS Suplpy Chain	Probable	Grave	9	Crear plan de acción	Verificación entidades y atributos desarrollados en el modelo ER con la documentación.
“Alucinaciones” de Amazon Bedrock	Altamente probable	Moderado	8	Crear plan de acción	Evaluar otro modelo de LLM y ajustar el “prompt” para generar respuestas más concisas y precisas.
Tiempo excesivo de respuesta de Amazon Bedrock	Altamente probable	Leve	4	Crear plan de acción	Implementar ingeniería de prompt
Bugs en “deployment” de servicios AWS	Altamente probable	Leve	4	Crear plan de acción	Utilizar Amazon CloudWatch logs para identificar errores y eliminarlos
Retrasos en el desarrollo debido a la complejidad	Probable	Leve	3	Crear plan de acción	Desarrollar un plan detallado, dividir el trabajo en módulos y establecer hitos claros.

Tabla 6: Matriz de riesgos y mitigaciones

Metodologías

Como metodología de trabajo para el proyecto se utilizó una metodología de cascada para la elaboración y desarrollo de las etapas generales. Sin embargo, para la etapa general “Desarrollo de la solución” se utilizó la metodología Scrum. De esta manera, se definieron las siguientes etapas generales:

Etapa	Detalle etapa	Tareas
0	Documentación	Z
1	Levantamiento del problema y requerimientos	A-D
2	Definición de la solución	E-H
3	Desarrollo de la solución	I-M
4	Implementación solución	N-O
5	Evaluación de la solución	P

Tabla 7: etapas generales definidas para el proyecto. Notar que en la etapa número 3 fue utilizada la metodología Scrum

La utilización de una metodología de cascada tuvo como objetivo elaborar un plan de trabajo altamente específico y estricto en cuanto al tiempo necesario para cada una de las etapas y su cumplimiento. Lo anterior debido al poco tiempo disponible para la elaboración del proyecto ingenieril, entregando así la posibilidad de elaborar una planificación sumamente cuidadosa respecto al tiempo. Además, se buscó generar un plan general de trabajo que vaya de la mano con los requisitos de la asignatura "Pasantía full-time", cumpliendo así con los plazos y avances definidos en el syllabus de la asignatura.

La metodología de trabajo de cascada al ser lineal no es la más apropiada para contextos de desarrollo de software¹⁸. Es por lo anterior que una vez clara la solución y el problema que se iba a abordar en el proyecto, se decidió trabajar con una metodología de trabajo ágil tipo Scrum, ajustada al equipo del proyecto (supervisor y alumno). Específicamente, en la etapa general “Desarrollo de la solución” se trabajó con la metodología mencionada. El objetivo principal de esta metodología es tener un progreso iterativo enfocado en la entrega rápida de componentes de nuestra solución. Se realizaron “sprints” de trabajo cortos de 5 días para el cumplimiento de tareas específicas de esta etapa, con reuniones de

¹⁸ Hoory, L., & Bottorff, C. (2022, Agosto 10). *Agile vs. Waterfall Methodology*. Forbes. Retrieved Diciembre 23, 2023, from <https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/>

evaluación junto con el Product Owner (alumno y supervisor) todos los lunes durante el periodo. El equipo de desarrollo y Scrum Master fue el alumno. Las tareas específicas del desarrollo de solución corresponden al "Product Backlog" (todas las tareas necesarias que se necesitan realizar para el cumplimiento del proyecto).

Planificación

El proyecto consta de cinco etapas generales además de una etapa de documentación (etapa 0). Como se mencionó en la sección de metodología, cada una de las etapas fue trabajada de manera secuencial según la metodología de cascada. Sin embargo, para las tareas específicas de la etapa 3 (correspondiente a la etapa "Desarrollo de la solución") se utilizó la metodología de trabajo tipo "Scrum". El desglose de las etapas generales y tareas (etapas específicas) se encuentra disponible en el Anexo 2 y 3, respectivamente.

ETAPAS	Tarea	Semana																	
		21-08	28-08	04-09	11-09	18-09	25-09	02-10	09-10	16-10	23-10	30-10	06-11	13-11	20-11	27-11	04-12	11-12	
0	Z																		
1	A																		
	B																		
	C																		
	D																		
2	E																		
	F																		
	G																		
	H																		
3	I																		
	J																		
	K																		
	L																		
	M																		
4	N																		
	O																		
5	P																		

Imagen 2: Carta Gantt de 17 semanas de trabajo para la finalización del proyecto.

Evaluación Económica

Para realizar una evaluación económica del proyecto, se utilizó el criterio de Valor Actual Neto (VAN) con diferentes combinaciones de posibilidades respecto a beneficios y tasas de descuento, evaluando así el VAN para 6 combinaciones de posibles casos con el fin de analizar la sensibilidad. Se utilizó un horizonte de 5 trimestres para el análisis: último trimestre del 2023 y todo el año 2024.

Para calcular los beneficios se calculó lo siguiente.

1. Costos del proyecto o inversiones: este valor corresponde al valor hora del pasante que desarrollará el proyecto multiplicado por la cantidad de horas invertidas mensualmente por el alumno en su desarrollo. De esta manera, se obtuvo un costo de CLP\$300.000 mensuales (detalle en Anexo 5). Así, el trimestre número 4 del 2023 se invirtieron CLP\$900.000 (tres meses de trabajo del alumno) y en el primer trimestre del 2024 se invirtieron CLP\$600.000 (el alumno tiene contrato hasta febrero). Respecto a los costos de infraestructura, estos son despreciables debido a que cada SA cuenta con cuentas en la consola de AWS que no tienen costos reales asociados.
2. Beneficios del proyecto: se utilizó como beneficio de este proyecto la monetización del ahorro de horas que tendría la implementación de este (6 horas que los SA podrían dedicar a otra cosa por cada demostración desarrollada). Para lo anterior, se calculó el valor de HH de un SA ponderado según la cantidad de SAs de un nivel específico y su respectivo sueldo estimado en Chile. Se obtuvo un valor de HH ponderado de un SA de \$20.370 (para más detalle revisar Anexo 6).

Por otro lado, se elaboraron tres casos de sensibilidad en base a la cantidad de demostraciones realizadas por cada trimestre: 3 (pesimista, ahorro de 18 horas por trimestre), 6 (neutral, ahorro de 36 horas por trimestre), 12 (optimista, ahorro de 72 horas por trimestre). Los resultados se encuentran en la gráfica de la Tabla 8.

	Caso Pesimista	Caso Neutral	Caso Optimista
Demostraciones AWS Supply Chain por trimestre	3	6	12
Horas ahorradas por trimestre	18	36	72
Beneficio económico trimestral por ahorro de horas	\$ 366.667	\$ 733.333	\$ 1.466.667

Tabla 8: desglose beneficios por cada trimestre.

Una vez definidas las inversiones y beneficios del proyecto, se elaboró un flujo en base a los tres casos mencionados (revisar gráfica de la Tabla 9).

Año	Trimestre	Inversión (costo)	Utilidad: caso pesimista	Utilidad: caso neutral	Utilidad: caso optimista
2023	4	-\$ 900.000	-\$ 533.333	-\$ 166.667	\$ 566.667
2024	1	-\$ 600.000	-\$ 233.333	\$ 133.333	\$ 866.667
	2	0	\$ 366.667	\$ 733.333	\$ 1.466.667
	3	0	\$ 366.667	\$ 733.333	\$ 1.466.667
	4	0	\$ 366.667	\$ 733.333	\$ 1.466.667

Tabla 9: flujo de cada caso con horizonte de 5 trimestres

Respecto a la tasa de descuento, se utilizaron 3 posibles casos de retorno anual: 10%, 15%, y 30% (2.5%, 3.8%, y 7.5% tasa trimestral, respectivamente).

Con los flujos de la Tabla 9 y las tasas de descuento trimestrales mencionadas anteriormente, se desarrolló el análisis de VAN de cada una de las 9 combinaciones de casos (tres tasas de descuento y tres casos de beneficios). De esta manera se obtuvo no existe ninguna combinación de cantidad de mostraciones y tasas que genere un VAN negativo. Por ende, el proyecto es viable económicamente.

<u>VAN</u>	Tasa trimestral 2,5%	Tasa trimestral 3,8%	Tasa trimestral 7,5%
Beneficios caso pesimista	\$254.333	\$218.982	\$127.082
Beneficios caso neutral	\$1.957.803	\$1.862.845	\$1.610.573
Beneficios caso optimista	\$5.364.744	\$5.150.570	\$4.577.556

Tabla 10: Resultado de análisis VPN para los 6 casos evaluados

Para verificar la decisión, se calculó la tasa de descuento utilizando el método CAPM ¹⁹. De esta forma, se obtuvo una tasa de descuento anual del 13.92% o del 3.48% trimestral (revisar Anexo 7). Luego, podemos confirmar que el proyecto es viable económicamente.

Medidas de desempeño

Las medidas de desempeño seleccionadas para la evaluación del proyecto según los objetivos definidos son las siguientes:

1. Tiempo promedio que toma el desarrollo de una demostración en base a una muestra N:

$$\text{Tiempo promedio de desarrollo de una demostración} = \frac{\sum_{i=1}^N (t_i + P)}{N} [\text{horas}]$$

t_i : tiempo que toma la generación de datos

P : Tiempo promedio que toma el desarrollo de material complementario según SAs

N : Cantidad de demostraciones desarrolladas

Ecuación (1)

Para definir el valor de “P” se realizó una encuesta a los SAs de Chile respecto al tiempo que les toma el desarrollo de material de presentación y la gestión para poder realizar una demostración (ejemplo: documentos, FAQs (“Frequently Asked Questions”), desarrollo Power Points, solicitud salas, etc.). El resultado fue un valor de 4 horas (revisar Anexo 9).

2. Medir el nivel de satisfacción de los SAs respecto a la solución desarrollada: mediante una encuesta a los SAs en Chile se cuantificar el nivel de satisfacción respecto a la efectividad de la solución. La encuesta está disponible en el Anexo 13.

3. Modelo entidad relación de AWS Supply Chain:

$$\text{Porcentaje entidades requeridas usadas} = \frac{N^{\circ} \text{ de entidades usadas}}{18} \times 100 \times \alpha [\%]$$

$$\alpha = \begin{cases} 1, & \text{si el modelo es aceptado por AWS Supply Chain} \\ 0, & \text{en otro caso} \end{cases}$$

Ecuación (2)

¹⁹ Kvilhaug, S. (2023, May 24). *What Is the Capital Asset Pricing Model (CAPM)?* Investopedia. Retrieved October 18, 2023, from <https://www.investopedia.com/terms/c/capm.asp>

Esta medida busca identificar si abarcamos el total de entidades necesarias para demostrar las características de planificación de demanda y recomendaciones de AWS Supply Chain. Además, contempla si el modelo (entidades y atributos) son aceptados por el servicio.

4. Países contingentes a la solución:

$$\text{Porcentaje de cobertura} = \frac{\text{Nº de países contingentes abarcados}}{4} \times 100 [\%]$$

Ecuación (3)

Esta medida busca medir si estamos cumpliendo con el objetivo de desarrollar una solución contingente a los 4 previamente definidos: Argentina, Chile, Colombia, y Brasil.

5. Avance del desarrollo de la solución:

$$\text{Porcentaje de avance} = \frac{\text{Nº de hitos alcanzados}}{\text{Nº total de hitos planificados}} \times 100 [\%]$$

$$\text{Hitos planificados} = \text{Tareas I, J, K, L, M, N O}$$

Ecuación (4)

Esta medida permite definir el progreso y el cumplimiento de las tareas específicas de las etapas de desarrollo de solución e implementación (revisar Anexo 2) dentro de los márgenes de tiempo de la pasantía.

Desarrollo del proyecto

Cómo se mencionó en la sección de metodología y planificación, el desarrollo de la solución fue mediante el uso de la metodología tipo Scrum. En total se realizaron 7 “sprints” de 5 días laborables cada uno.

Todo lo desarrollado en el proyecto se encuentra disponible en el link del repositorio del Anexo 10.

Diseño y desarrollo de modelo de entidad relación para AWS Supply Chain

Para que la solución propuesta sea útil, debemos cumplir con que el esquema de base de datos que vamos a generar sea coherente con los requerimientos establecidos por AWS Supply Chain. Además de lo anterior, estos datos deben seguir la lógica de una cadena de suministro teórica, que involucre a las entidades necesarias para demostrar el funcionamiento del servicio respecto a la planificación de demanda²⁰ y recomendaciones²¹.

Dicho esto, el primer paso en el desarrollo de la solución fue la creación de un esquema de cadena de suministro teórico, resumido en el siguiente diagrama:

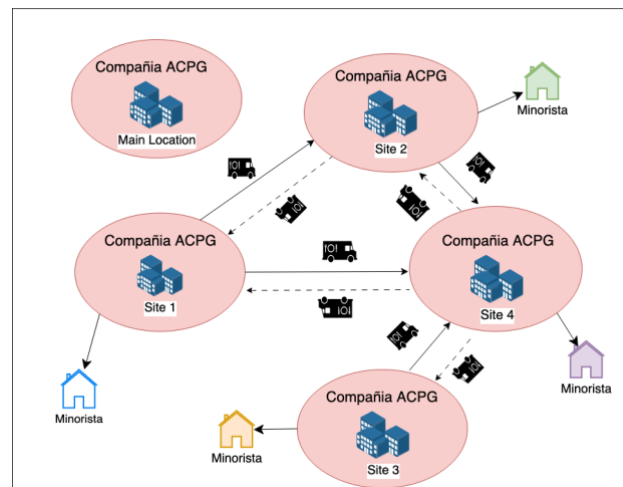


Imagen 3: ejemplo de esquema de cadena de suministro de 3 sitios y una localidad principal. Solución implementada tiene 6 y una localidad de “main location” (localidad principal).

²⁰ AWS Supply Chain. (2022). *Data entity columns required by Demand Planning - AWS Supply Chain*. AWS Documentation. Retrieved Diciembre 01, 2023, from https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/required_entities.html

²¹ AWS Supply Chain. (2022). *Data entity columns required by Insights - AWS Supply Chain*. AWS Documentation. Retrieved Diciembre 01, 2023, from <https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/entities-insights.html>

En resumen, se simula una cadena de suministro que sigue la lógica de una industria “Customer Packaged Goods” (bienes de consumo envasado) que consta de datos maestros (datos fijos. Por ejemplo: cantidad de localidades, productos, políticas de inventario, direcciones de los sitios, etc.) y transaccionales (que representan una transacción y, por ende, tienen una fecha de suceso. Por ejemplo: ordenes de productos, productos recibidos, productos pedidos, etc.). En ella se realizan envíos de ordenes por lote de múltiples productos entre los distintos sitios de la empresa según una política de transporte²² específica (definida aleatoriamente). Las cantidades pedidas, enviadas y confirmadas de cada producto en un lote sigue una distribución normal con parámetros específicos por producto. Del mismo modo, los productos salientes hacia los minoristas siguen una distribución normal de media y desviación estándar específica por cada producto. Además, las políticas de inventario del modelo de cadena de suministro son definidas tomando en cuenta los parámetros (media y desviación estándar) de demanda y “lead time” (retraso de la orden). Los envíos son realizados por una empresa externa.

Una vez con el esquema de cadena de suministro definido, se procedió a representarlo en un modelo de entidad relación siguiendo los requerimientos de entidades y atributos del servicio en cuestión. Así, se obtuvo el siguiente modelo:

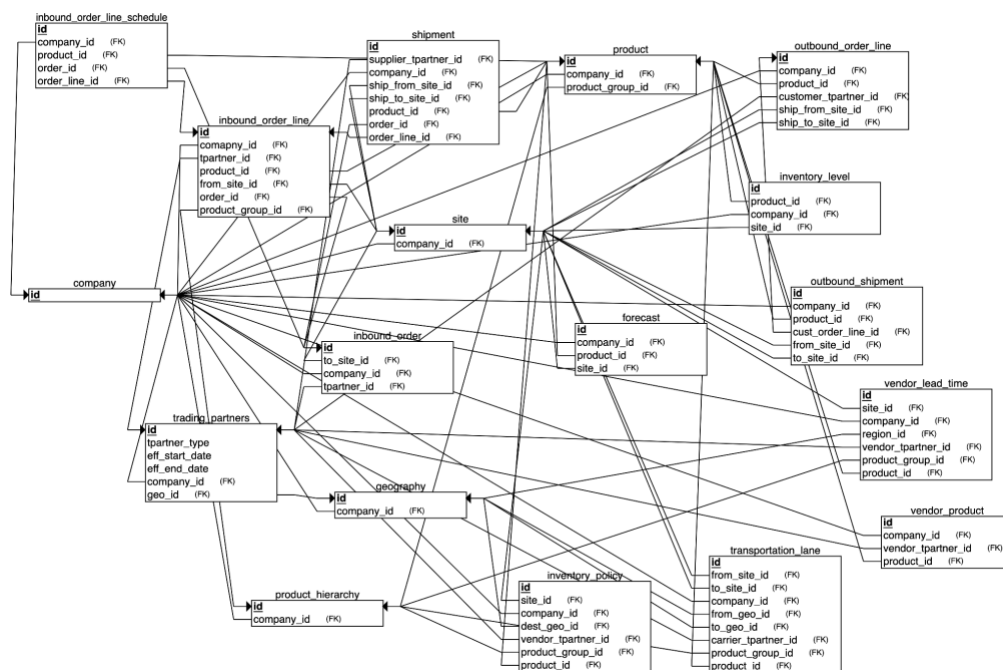


Imagen 4: representación del esquema de cadena de suministro en entidad relación

²² Política de transporte: indica cuales de los productos se pueden enviar desde un sitio “X” hacia un sitio “Y”. No necesariamente se pueden enviar todos los productos desde y hacia todos los sitios.

Es fundamental mencionar que para tener una mejor visualización del modelo solamente se muestran las llaves primarias y secundarias de cada entidad (revisar Anexo 8 para más detalle), sin embargo, cada una contiene más atributos (disponibles en la documentación de AWS Supply Chain²³). Cada una de estas entidades será poblada con datos sintéticos y guardada en formato “csv” en Amazon S3 para posteriormente poder realizar la ingesta hacia AWS Supply Chain. Lo anterior, será realizado por el algoritmo desarrollado en la tarea de “Diseño y desarrollo back-end”.

Diseño y desarrollo front-end

El objetivo de la plataforma es que el usuario logre seleccionar tanto la industria (mostrando así productos contingentes a la industria) y el país (Colombia, Argentina, Chile o Brasil), para así poder generar datos para el servicio según el caso específico del cliente.

Para el desarrollo se utilizó el framework Next.js con Tailwind. Fue alojado en AWS Amplify, servicio de AWS destinado a permitir crear y alojar plataformas web full-stack de manera fácil y expedita. Desde el front-end, podemos generar los datos cuando presionamos un botón con el icono de “play” y podemos chequear el estado de estos cuando presionamos el botón “Check data”, que modifica el icono de la columna “Ready” a un ticket verde en caso de que el “csv” de la entidad ya se haya creado (revisar Imagen 5 y 6). Una vez que los datos han sido generados, podemos copiar el texto de la columna “S3 URL” y entregárselo a AWS Supply Chain para poder realizar la integración de los datos.

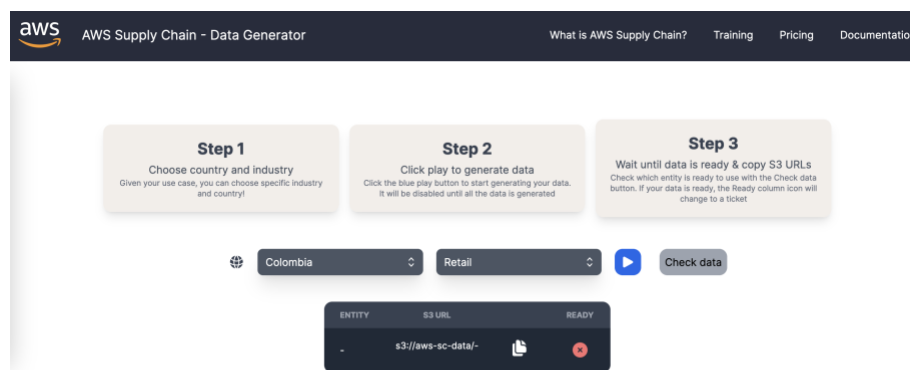


Imagen 5: pág. de inicio (botón de “play” activo, el “Check data” inactivo y tabla de S3 urls incompleta)

²³ AWS. (2022, Noviembre 29). *Data entities supported in AWS Supply Chain - AWS Supply Chain*. AWS Documentation. Retrieved Octubre 01, 2023, from <https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/data-model.html>

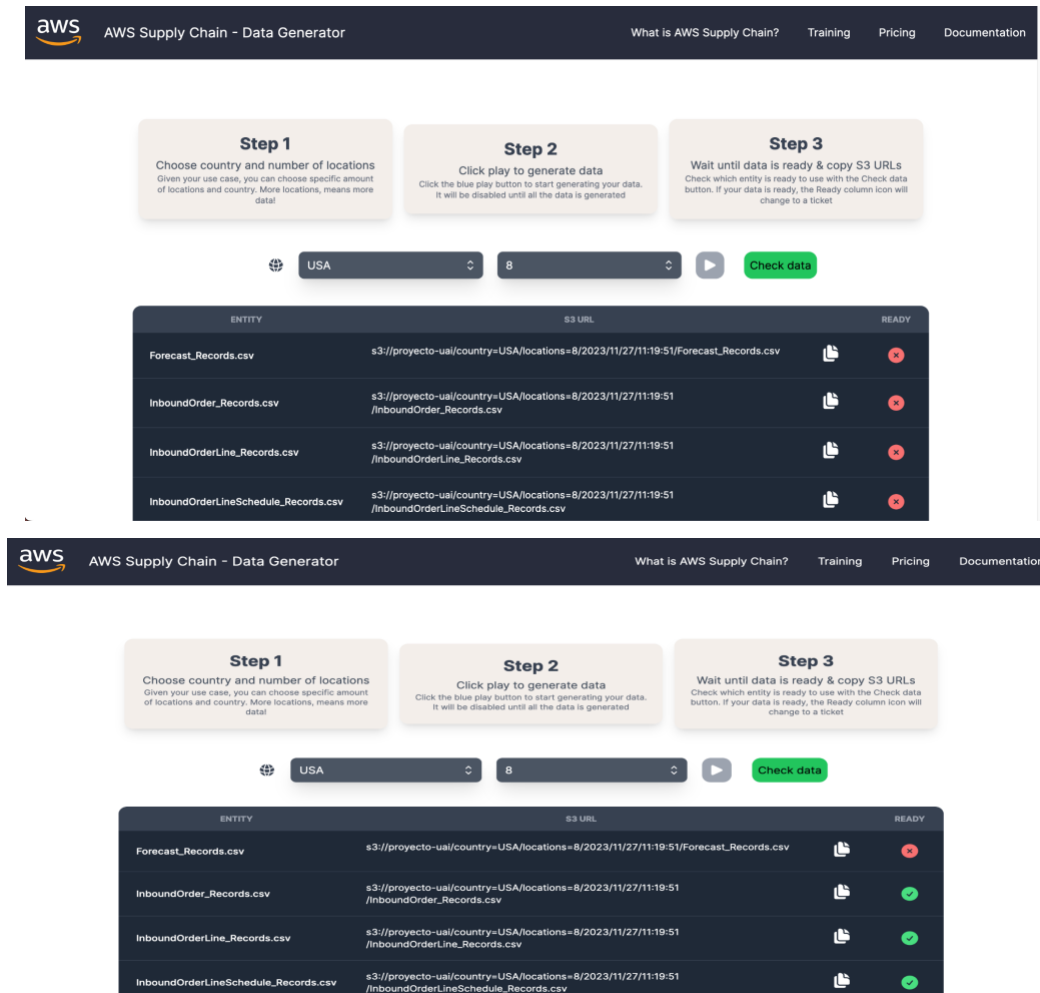


Imagen 6 y 7: La primera muestra una el resultado una vez que se presiona el botón de “play”, mientras que la segunda muestra el cambio del icono de la columna “Ready” una vez que se generó la data (tres entidades ya generadas)

Desarrollo back-end

Para el back-end de la plataforma se utilizaron servicios serverless de AWS, desarrollando así una arquitectura basada en eventos. Los servicios de AWS utilizados fueron:

1. AWS Lambda: servicio de cómputo sin servidor que permite ejecutar código sin provisionar ni administrar servidores.
2. Amazon API-Gateway: permite crear, publicar, mantener, monitorear y asegurar APIs a cualquier escala.

3. AWS Step Function: permite coordinar componentes de aplicaciones serverless usando flujos de trabajo. Los pasos individuales en un flujo de trabajo se ejecutan en forma secuencial o en paralelo utilizando funciones Lambda.
4. Amazon S3: servicio de almacenamiento de objetos en la nube, escalable y de alto rendimiento, que permite almacenar y recuperar cualquier cantidad de datos en cualquier momento y desde cualquier lugar.
5. Amazon Bedrock: servicio que ofrece una selección de modelos fundacionales (FMs) de alto rendimiento de las principales empresas de IA en una sola API, junto con un amplio conjunto de capacidades necesarias para crear aplicaciones de GenIA.



AWS Lambda

Amazon
API-GatewayAWS Step
Function

Amazon S3

Amazon
Bedrock

Iconos 1, 2, 3, 4 y 5: servicios AWS utilizados.

La arquitectura final del back-end se encuentra en la gráfica de la Arquitectura 4.

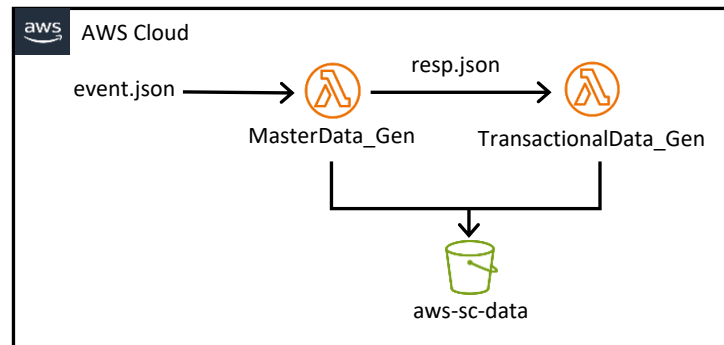
Desarrollo algoritmos de generación de datos con AWS Lambda

Para los algoritmos de generación de datos, se utilizó AWS Lambda. Se desarrollaron dos funciones en Python 3.9 que generan archivos “csv” para cada una de las entidades mencionadas en el Anexo 8, siguiendo la lógica del esquema de cadena de suministro definido previamente (gráfica de la Imagen 3 y 4). Las funciones son las siguientes:

1. AWS_SC_MasterData_gen: tiene como objetivo generar datos de las entidades maestras requeridas (revisar Anexo 8). Esta función requiere que se le entregue un evento en formato json que contenga los siguientes atributos: “datetime”, “country”, “industry”, “products” y “locations” (para más detalle ingresar al repositorio disponible en Anexo10: archivo en /lambdas/AWS_SC_MasterData_gen/test-payload.json). Además, esta función genera una respuesta en formato json que será el evento entregado a la siguiente función (“resp.json” del diagrama Arquitectura 1).
2. AWS_SC_TransactionalData_gen: tiene como objetivo generar los datos de las entidades transaccionales requeridas (revisar Anexo 8). Esta función requiere que se le entregue un evento en formato json que contenga los siguientes atributos: “datetime”, “country”, “industry”,

“products” y “sites” (para más detalle ingresar al repositorio disponible en Anexo 10: archivo en /lambdas/AWS_SC_TransactionalData_gen/test-payload.json). Esta función depende de la primera: requiere los mismos datos, pero con un pre procesamiento específico realizado en la primera función.

Estas funciones Lambda generan los datos y los guardan en formato “csv” dentro de un “bucket” de Amazon S3 llamado "aws-sc-data", servicio con el que podemos realizar la integración hacia AWS Supply Chain.



Arquitectura 1: generación de datos con AWS Lambda y almacenamiento en S3

Generación de datos base con Amazon Bedrock

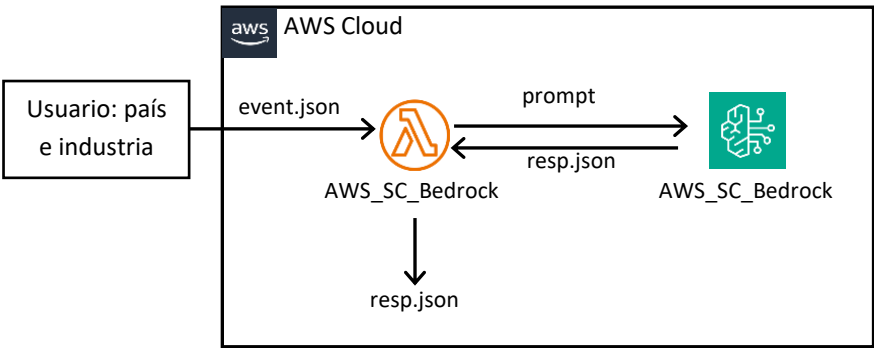
Se utilizó Amazon Bedrock como herramienta de GenIA para generar los datos del evento que se debe entregar a las funciones Lambda mencionadas en la sección anterior (“event.json” en la figura Arquitectura 1). Esto lo hace en base a la industria y país requerido, parámetros definidos por el usuario en el front-end de la plataforma.

Para implementar Amazon Bedrock, se utilizó una función Lambda llamada “AWS_SC_Bedrock”. En ella, se le entregan 3 prompt²⁴ (más información en Anexo 12) al algoritmo LLM (específicamente, se utilizó el modelo “Claude v2” de Anthropic²⁵) para generar los productos contingentes a la industria, localidades contingentes al país y los parámetros de inventario y demanda de cada producto generado. Posteriormente, los datos mencionados se reformatean para ser la respuesta de la función (“resp.json”

²⁴ “Prompt”: entradas o consultas que un usuario o un programa proporciona a una LLM para obtener una respuesta específica del modelo.

²⁵ Anthropic. (2023, Julio 11). *Claude 2*. Anthropic. Retrieved Diciembre 03, 2023, from <https://www.anthropic.com/index/claude-2>

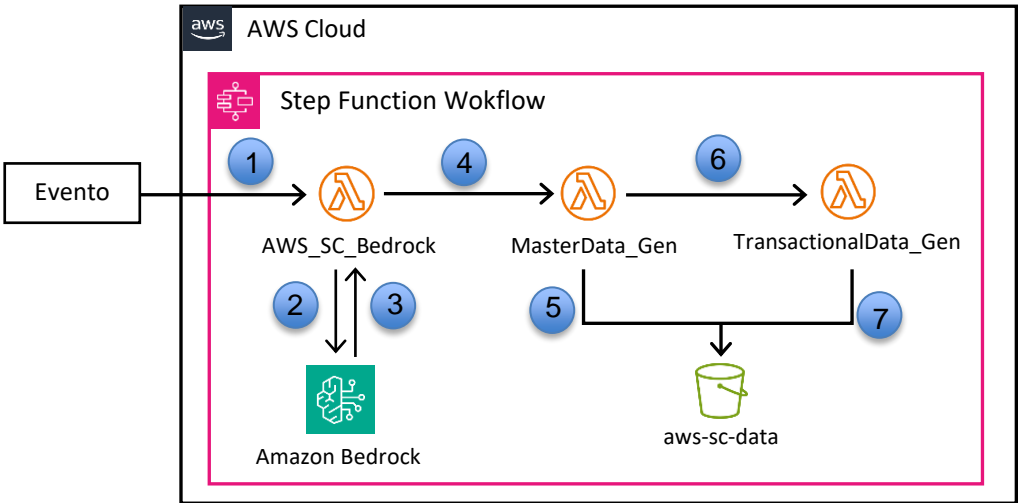
de la función Lambda en diagrama Arquitectura 2) Esta respuesta contiene los atributos de “datetime”, “country”, “industry”, “products” y “locations”, todos requeridos por las funciones “AWS_SC_TransactionalData_gen” y “AWS_SC_MasterData_gen”.



Arquitectura 2: generación de datos con Amazon Bedrock y AWS Lambda

Integración con AWS Step Function y Amazon API Gateway

En primer lugar, para integrar las 3 funciones Lambda mencionadas anteriormente, se utilizó el servicio AWS Step Function. De esta manera, podemos generar una secuencia específica de ejecución donde primero ejecutamos la función “AWS_SC_Bedrock”, luego “AWS_SC_MasterData_Gen”, y por último la función “AWS_SC_TransactionalData_Gen”. Las funciones se ejecutan una vez que ocurra cierto evento (“Evento” en diagrama Arquitectura 3). En este caso, el evento debe ser accionado por el botón azul de “play” del front-end (revisar Imagen 5).

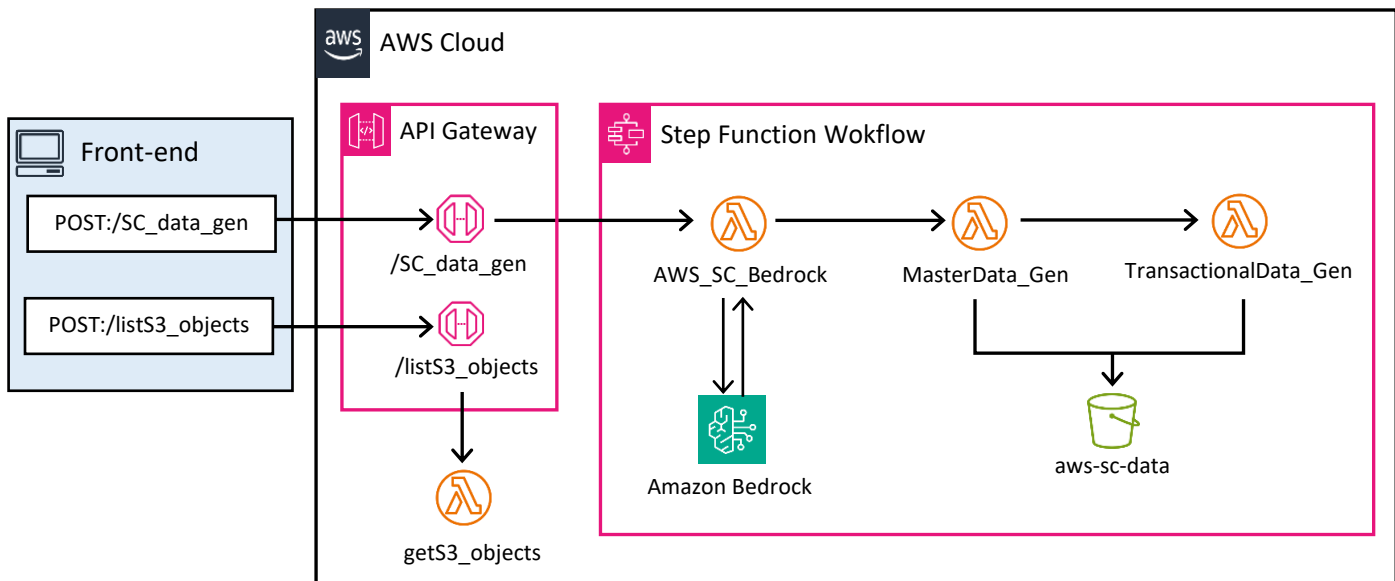


Arquitectura 3: integración funciones Lambda utilizando AWS Step Function. Los números representan el orden de ejecución de los procesos involucrados.

En según lugar, para integrar el front-end con el back-end se utilizó el servicio AWS API Gateway. Específicamente, se crearon dos rutas (endpoints):

1. `/SC_data_gen`: cuando el usuario presiona el botón con el icono "play" en la plataforma (botón azul de la Imagen 5), se hace un "POST request" a este endpoint. Este suceso es considerado como el evento que gatilla la ejecución del Step Function, ejecutando así la secuencia definida en el diagrama de la Arquitectura 3.
2. `/listS3_objects`: cuando el usuario presiona el botón de "Check data" (botón verde de la Imagen 6), se hace un "POST request" a esta ruta. Una vez que llega el "request" a la API, se ejecuta una función Lambda llamada "getS3_objects". Esta función tiene como objetivo retornar cuál de las entidades ya se encuentra disponible en el bucket "aws-sc-data" de Amazon S3. Así, somos capaces de actualizar el estado de cada entidad y mostrarlo al usuario mediante el cambio del icono de la columna "Ready" de la página (revisar Imagen 7).

Así, la arquitectura completa del back-end quedó de la siguiente forma:



Arquitectura 4: arquitectura final del back-end.

Implementación

Por último, se realizó la implementación de la plataforma utilizando infraestructura como código (IaC) y documentación. Se generaron archivos en formato json que contienen todas las características de la

arquitectura. Estos archivos pueden ser entregados al servicio AWS CloudFormation²⁶ para levantar todos los servicios que se observan en la gráfica de la Arquitectura 4. Además, se elaboró un instructivo paso a paso para levantar la arquitectura en la cuenta de cada uno de los SAs y así poder generar los datos para la demostración de AWS Supply Chain. Todo lo mencionado se encuentra en el repositorio del Anexo 10.

²⁶ AWS CloudFormation es un servicio que ofrece a desarrolladores y empresas una manera sencilla de crear una colección de recursos de AWS. También ayuda a aprovisionarlos y administrarlos de forma ordenada y predecible.

Resultados cualitativos y cuantitativos

Medida de desempeño 1

En primer lugar, para evaluar esta medida de desempeño se tuvo que definir el valor de “P” (mencionado en la Ecuación 1 de la sección “Medidas de desempeño”). Para ello, se encuestó a los 12 SAs de Chile, obteniendo así un valor promedio de 4.0 horas (revisar Anexo 9).

Una vez definido, se realizaron 5 pruebas de demostraciones para la generación de datos y se obtuvieron los resultados de la Tabla 11. Para el cálculo del tiempo que toma la ejecución del algoritmo de generación de datos, se utilizó Amazon CloudWatch para evaluar el tiempo que toma la ejecución del Step Function en las 5 iteraciones mencionadas. Los resultados se muestran a continuación.

	Demostración 1	Demostración 2	Demostración 3	Demostración 4	Demostración 5
Fecha	16-11-23	20-11-23	22-11-23	24-11-23	27-11-23
Hora inicio	16:44:34	11:17:54	15:21:42	08:56:48	09:38:45
Hora termino	16:50:01	11:22:45	15:26:43	09:02:21	09:44:43
Tiempo (min)	00:05:27	00:04:51	00:05:01	00:05:33	00:05:58
ti (hrs)	0,09	0,08	0,08	0,09	0,10
P	4,0	4,0	4,0	4,0	4,0
ti + P	4,07	4,06	4,06	4,07	4,08

Tabla 11: resultados de tiempo que toma la generación de datos.

De esta forma, se obtuvo un promedio de 5:04 minutos o 0.09 horas de duración en la generación de datos. Así, el resultado del tiempo promedio de desarrollo de una demostración luego de implementar el proyecto es de 4.1 horas, superando así el valor definido por el objetivo específico número 1. En otras palabras, el tiempo requerido por un SA para la elaboración de una demostración disminuyó un 69%.

Medida de desempeño 2

Respecto a la medida de desempeño cualitativa, el resultado de la encuesta arrojó que existe una satisfacción general del 86%, una satisfacción del 82% respecto a la disminución de esfuerzo en el desarrollo de una demostración del servicio, una satisfacción del 70% respecto a la claridad del caso de

uso e industria utilizada, y una satisfacción del 85% respecto a la demostración de las funcionalidades con los datos generados. Los resultados de la encuesta se encuentran disponibles en Anexo 13.

Medida de desempeño 3

Cómo ya se ha mencionado anteriormente, la plataforma genera 18 entidades diferentes: 10 que corresponden a datos maestros y 8 correspondientes a datos transaccionales, almacenándolas todas en un bucket específico de Amazon S3. Todas ellas fueron desarrolladas minuciosamente según la documentación del servicio. Las entidades generadas son aceptadas por el servicio AWS Supply Chain, independiente de los parámetros definidos por el usuario (país e industria). Así, obtenemos un valor de “α” de 1 y cumpliendo con que el valor de esta métrica sea del 100%.

$$\text{Porcentaje de entidades requeridas utilizadas} = \frac{18}{18} \times 100 \times 1 [\%] = 100 \%$$

Medida de desempeño 4

El front-end de la plataforma desarrollada permite seleccionar entre 4 países para generar los datos: Argentina, Brasil, Chile, y Colombia. Sin embargo, al estar utilizando GenAI para la generación de los datos, podríamos pedir a Amazon Bedrock datos contingentes a cualquier país e industria del mundo. De esta forma, tenemos un valor del 100% en la métrica de “Porcentaje de cobertura”.

$$\text{Porcentaje de cobertura} = \frac{4}{4} \times 100 [\%] = 100\%$$

Medida de desempeño 5

Cómo se logró observar en la sección “Desarrollo de la solución”, se realizaron las 7 tareas específicas de esta etapa. De esta forma, cumplimos con la completitud del avance de la solución en los márgenes de tiempo correspondientes, obteniendo un 100% en la métrica “Porcentaje de Avance” a la fecha 11 de diciembre del 2023.

$$\text{Porcentaje de avance} = \frac{7}{7} \times 100 [\%] = 100\%$$

$$\text{Tareas finalizadas} = I, J, K, L, M, N, O$$

Conclusión

A modo de conclusión, el proceso de venta de los servicios de AWS es un proceso complejo que involucra múltiples cargos dentro de la compañía. El AWS Account Manager acompaña al cliente en toda su vida dentro de AWS, mientras que los Solutions Architect tienen entre sus tareas demostrar el funcionamiento y los beneficios de los servicios de AWS a los clientes, siendo los responsables de validar el aspecto técnico de las oportunidades de venta. Este punto era un dolor significativo en las oportunidades del servicio AWS Supply Chain, que busca entregar inteligencia logística similar a la utilizada por Amazon.com. Así, el tiempo y esfuerzo invertido por los SAs en estas demostraciones es sumamente excesivo al representar un 20% del total de tiempo de este tipo de tareas.

Para mitigar los dolores identificados, se propusieron tres soluciones con el objetivo de mejorar el proceso de venta del servicio AWS Supply Chain, para en 3 meses disminuir un 50% el tiempo invertido por un SA en las oportunidades en “Technical Validation” del servicio. Luego de un análisis de las tres soluciones en base a criterios ponderados, se decidió desarrollar una plataforma de generación de datos que permite desarrollar demostraciones según el caso de uso específico del cliente, lo anterior gracias a la implementación de Inteligencia Artificial Generativa con el servicio de AWS llamado Amazon Bedrock.

Esta plataforma web sigue una arquitectura serverless basada en eventos, utilizando servicios como AWS Lambda, Amazon S3, Amazon API-Gateway, etc. El objetivo de ella fue disminuir el tiempo invertido por un SA en la elaboración de datos para demostrar el servicio a un potencial cliente, disminuyendo así un 69% el tiempo que le toma y obteniendo un nivel de satisfacción general del 82%. Además, la solución desarrollada cumple con generar los datos y entidades necesarias para demostrar las características de planificación de demanda y recomendaciones de AWS Supply Chain. Asimismo, la plataforma logra generar datos contingentes a los 4 países seleccionados (Argentina, Brasil, Chile, y Colombia) y a la industria del cliente, siendo así una solución fiel al principio de “obsesión por el cliente” de Amazon.

Por último, el proyecto fue desarrollado en su totalidad dentro del tiempo definido en la planificación. Así, se logró cumplir con todos los objetivos específicos del proyecto y, por consecuencia, el objetivo general.

Anexos

1. Encuesta: [Encuesta_SAs](#)
2. Detalles de las tareas específicas del proyecto y de la carta Gantt:

ETAPA	TAREA	DETALLE TAREA
0	Z	Documentación
1	A	Entendimiento del negocio y AWS Supply Chain
	B	Análisis interno de brechas y dolores
	C	Definición oportunidad
	D	Definición objetivos y KPIs
2	E	Análisis e investigación estado del arte
	F	Propuestas de solución
	G	Análisis de la solución y definición servicios AWS a usar
	H	Especificaciones técnicas solución
3	I	Diseño y desarrollo esquema de base de datos relacional según requisitos de AWS Supply Chain
	J	Diseño y desarrollo front-end interactivo para generación de datos
	K	Desarrollo back-end de la aplicación
	L	Desarrollo de generación de datos con Amazon Bedrock
	M	Integraciones
4	N	Implementación
	O	Test de implementación
5	P	Evaluación resultados cuantitativos y cualitativos

Imagen 8: detalle tareas específicas del proyecto

3. Detalle etapas generales del proyecto y de la carta Gantt

ETAPA	DETALLE ETAPA
0	Documentación
1	Levantamiento del problema
2	Definición de la solución
3	Desarrollo solución
4	Implementación solución
5	Análisis resultados

Imagen 9: detalle etapas generales del proyecto

4. Selección de las soluciones a desarrollar:

Con los criterios y soluciones propuestas, se entrevistó a 3 Solutions Architect miembros del equipo del alumno siguiendo la siguiente escala Likert para ponderar cada una de las opciones:

Puntuación	1	2	3	4	5
Escala Likert	Nada Satisfecho	Poco Satisfecho	Neutral	Muy Satisfecho	Totalmente Satisfecho

Imagen 10: escala Likert utilizada

Las respuestas se encuentran en la siguiente imagen:

Solucion 1					
Respuestas	Criterio				
	1	2	3	4	5
1	2	4	1	2	2
2	3	4	2	1	2
3	2	5	2	2	2
Promedio	2,3	4,3	1,7	1,7	2,0

Solucion 2					
Respuestas	Criterio				
	1	2	3	4	5
1	4	3	4	4	4
2	3	2	4	5	5
3	4	2	4	4	4
Promedio	3,7	2,3	4,0	4,3	4,3

Solucion 3					
Respuestas	Criterio				
	1	2	3	4	5
1	4	4	2	5	3
2	3	5	3	4	2
3	4	4	3	4	3
Promedio	3,7	4,3	2,7	4,3	2,7

Imagen 11: resultados de evaluación soluciones según los criterios definidos

Tomando en cuenta las ponderaciones definidas, se obtuvieron los siguientes resultados:

Solución	Criterio					Ponderación
	Nivel de complejidad para su uso	Nivel de confianza	Nivel de automatización	Tiempo dedicado SAs	Nivel de obsesión por el cliente	
1	2,3	4,3	1,7	1,7	2,0	2,3
2	3,7	2,3	4,0	4,3	4,3	4,2
3	3,7	4,3	2,7	4,3	2,7	3,6

Imagen 12: resultados ponderación de cada una de las soluciones

Luego, se seleccionó la solución con ponderación más alta.

5. Costo de la elaboración del proyecto:

Para el cálculo del valor hora del pasante se contempló el sueldo mensual de este y la cantidad de horas que dedica diariamente al desarrollo del proyecto (revisar gráfica de la Imagen 15).

	Valor HH
SA Intern	\$ 3.000

	HH diarias	hh por mes
HH dedicadas al proyecto	5	100

	Mensual
Costos Proyecto	\$ 300.000

Imagen 13, 14, y 15: valor hora practicante, cantidad de horas diarias dedicadas al proyecto y costo mensual del proyecto

6. Valor HH de un SA ponderado:

Para el cálculo del valor hora de un SA full-time en AWS se tomó en cuenta la proporción de SAs de diferentes niveles en base al total de ellos en equipos de Chile, y el sueldo aproximado según el nivel (L5, L6 y L7). De esta manera se obtuvo un valor hora SA ponderado de CLP\$20.370 los datos disponibles en la gráfica de la Imagen 16.

	Valor hh	Cantidad en Chile	%
SA L5	\$ 16.667	5	42%
SA L6	\$ 22.222	6	50%
SA L7	\$ 27.778	1	8%
Valor HH SA ponderado		\$ 20.370	

Imagen 16: Cantidad de SAs de cada nivel en Chile de equipos AWS SMB y DEEP ponderados. Resultado de valor HH SA ponderado

7. Cálculo tasa de descuento:

Para calcular la tasa de descuento anual y trimestral para el proyecto, se utilizó el modelo CAPM que toma en cuenta tres factores: tasa libre de riesgo, beta, y retorno promedio del mercado. La definición de los factores mencionados se realizó de la siguiente manera:

- Tasa libre de riesgo: se utilizó el retorno anual esperado de un bono de gobierno en Chile definida en un 7.15% el 18 de octubre del 2023 (Investing.com & TradingEconomics.com, n.d.)
- Beta: se utilizó el beta de la empresa Amazon.com ("AMZN") definido en la página de "Yahoo finance" (Yahoo! Finance, n.d.)
- Retorno esperado del mercado: para este valor se utilizó el retorno esperado del S&P500 debido a que es altamente representativo del mercado y Amazon.com está en él. De esta

forma, se definió en un valor en base al historial de 10 años de 12.84% anual (Velasquez, 2023).

De esta forma, se obtuvieron los resultados de la gráfica de la Imagen 17:

CAPM		
Tasa libre de riesgo (1 año)	Rf	7,15%
Beta	b	1,19
S&P500 retorno anual (10 años)	Rm	12,84%
Tasa de descuento (anual)		13,92%
Tasa de descuento (trimestral)		3,48%

Imagen 17: cálculo de la tasa de descuento para el proyecto

8. Entidades del esquema de base de datos desarrollado:

Para desarrollar el modelo de base de datos, se siguió la documentación de la herramienta AWS Supply Chain, seleccionando las entidades necesarias para poder demostrar las funcionalidades del servicio en cuestión. De esta forma, se desarrollaron las 18 entidades explicadas a continuación:

- a. Categoría "Inbound": conformada por 4 entidades transaccionales. Estas son "inbound_order" (información sobre los pedidos entrantes a las distintas ubicaciones de la empresa), "inbound_order_line" (información respecto a la orden en sí, como cantidad de producto, fecha esperada de la orden, localidad de origen, etc), "inbound_order_line_schedule" (información específica respecto a la planificación del envío de la orden: fecha, cantidad solicitada de cada producto, etc) y "shipment" (contiene información real del envío de la orden, no descripción de cada orden).

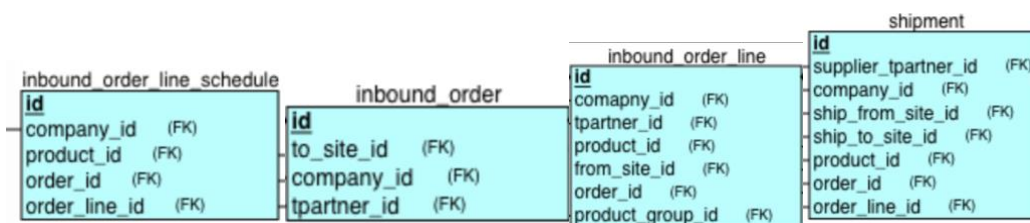


Imagen 18: entidades "inbound_order_line_schedule", "inbound_order", "inbound_order_line", y "shipment"

- b. Categoría "outbound fulfillment": conformado por 2 entidades transaccionales. Estas son "outbound_order_line" (información de órdenes originadas por la compañía y que salen desde sus localidades a otras fuera de su red) y "outbound_shipment" (información respecto a la orden saliente, como hacia dónde se dirige, desde donde se está enviando, fecha de envío, fecha esperada de envío, etc.).

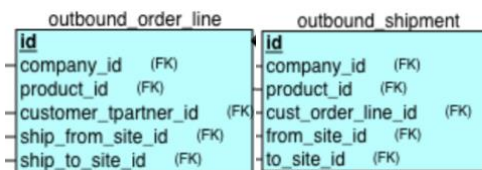


Imagen 19: entidades "outbound_order_line" y "outbound_shipment"

- c. Categoría "organization": conformado por 3 entidades no transaccionales. Estas son "company" (guarda información respecto de la compañía, como el nombre y dirección), "geography" (contiene información geográfica de la empresa, como el estado, provincia, ciudad, etc) y "trading_partner" (contiene información respecto a los "partners" que tienen relación con la empresa).

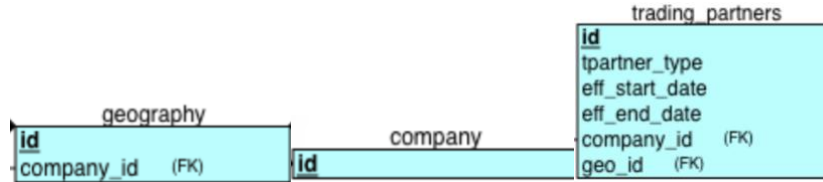


Imagen 20 entidades "geography", "company" y "trading_partners"

- d. Categoría "product": conformado por 2 entidades no transaccionales, "product" (información del producto, como descripción, precio, nombre, etc.) y "product_hierarchy" (describe las categorías de los productos).



Imagen 21: entidades "product_hierarchy" y "product".

- e. Categoría "planning": conformado por una única entidad no transaccional "inventory_policy". Esta define la política de inventario de la empresa, definiendo así factores como inventario de seguridad, inventario máximo y mínimo, etc.

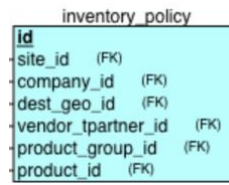


Imagen 22: entidad "inventory_policy"

- f. Categoría "network": conformado por 2 entidades no transaccionales. Estas son "site" (información respecto a las distintas localidades de la empresa) y "transportation_lane" (información de las líneas de transporte de la empresa, como distancia entre localidades, modo de transporte y tiempo).

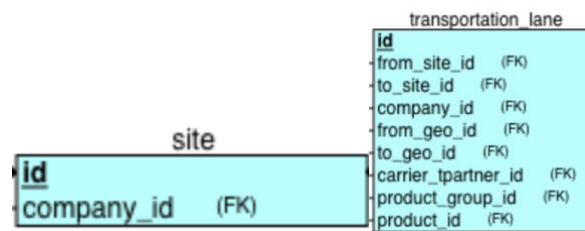


Imagen 23: entidades "site" y "transportation_lane".

- g. Categoría "vendor management": compuesto por 2 entidades no transaccionales. Estas son "vendor_product" (contiene información de producto por proveedor) y "vendor_lead_time" (contiene información actual y planeada del tiempo de atraso del proveedor).

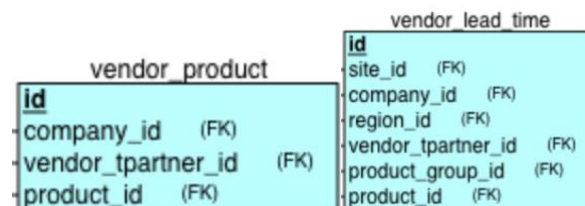


Imagen 24: entidades "vendor_product" y "vendor_lead_time".

- h. Categoría "forecast": conformado por una entidad transnacional denominada "forecast". Contiene información de pronóstico por cada localidad y producto.



Imagen 25: entidad "forecast".

- i. Categoría "inventory management": conformado por una única entidad transaccional llamada "inventory_level" que describe el estado actual del inventario en cada localidad de la compañía.

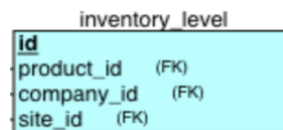


Imagen 26: entidad "inventory_level".

9. Definición del tiempo promedio que toma la generación de material de presentación para una demostración y la gestión de este:

Para definir este valor, se encuestó a los SAs de Chile. La pregunta fue la siguiente:

“Respecto a la preparación de una demostración con un cliente ¿Cuánto tiempo te toma el desarrollo de material no técnico? Ej: presentaciones, FAQs, documentos, petición de salas, gestión del catering, etc.”

Se entregaron las siguientes opciones: 1 hora, 2 horas, 3 horas, 4 horas, u otro (indicar el valor).

Los resultados se encuentran en la siguiente imagen:

Encuestado	Respuesta (hrs)
1	4
2	4
3	4,5
4	3,5
5	4
6	4,5
7	3,5
8	4,5
9	3,5
10	4
11	4,5
12	3,5
Promedio	4,0

Imagen 27: resultados encuesta

10. Guía de implementación: <https://github.com/cjdrago/UAI-aws/>

11. Desglose objetivo “SMART”:

- Específico: se busca impactar positivamente un eslabón específico del proceso de ventas (oportunidades en “Technical Validation”) de un único servicio de AWS.
- Medible: el tiempo que toma el desarrollo de una demostración es cuantificable y medible.
- Realizable: se cuenta con el equipo y conocimiento personal de tecnología y proceso industriales adecuados. Además, con el foco de automatizar el proceso, es factible reducir un 50% el tiempo que toma la realización de demostraciones en el tiempo definido.
- Relevante: la reducción del tiempo invertido por los SA en un cliente es relevante para la compañía al ser fundamental en la eficiencia del negocio: un SA tiene muchos clientes y buscamos que las oportunidades estén el menor tiempo posible en “Technical Validation” para avanzar a “Commit”.
- Tiempo: plazo específico de 3 meses desde el comienzo de la práctica profesional (16 de agosto del 2023).

12. Prompts de Amazon Bedrock:

i. Ejemplo prompt 1:

```
Context: Wine industry in Chile. The answer must be a json in one line: <response>[json]</response>. \n
Mention 8 products. Json properties: name, category (one word), description, company (fake), price (in
Chile), unit of measure, id (6 characters), currency (Chile currency).
```

ii. Ejemplo prompt 2:

```
Context: Wine industry in Chile. The answer must be a json in one line: <response>[json]</response>. \n
Mention 8 locations in cities of Chile. Json properties: address (street and number), city (name), id (8
characters), latitude (string), longitude (string), time_zone, postal_code (address postal code),
phone_number, country_code (Chile ISO Alpha-2 code), state_prov, main (set to true only for one location,
for the others must be set to false), tpartner (set to true only for one location different than the main one, for
the others must be set to false).
```

iii. Ejemplo prompt 3:

```
Context: Wine industry in Chile, 'Cabernet Sauvignon' product with price of 7000 CLP. \n The answer must
be a json in one line: <response>[json]</response>. \n Generate an inventory policy and vendor lead time for
'Cabernet Sauvignon' product in 6 different sales locations (location_id: IJKLMNOP, PQRSTUVWXYZ,
GHIJKLMO, PQRSTUVWXYZ, XYZABCDE, FGHIJKLM). Json properties: location_id, min_safety_stock (unit of
```

measure of 'bottle'), max_safety_stock (unit of measure of 'bottle'), planned_lead_time (in days), planned_lead_time_dev (in days), planned_lead_time_mean (in days).

13. Encuesta de evaluación cualitativa de satisfacción de la solución 1: esta encuesta se enfocó en evaluar el desempeño de la solución respecto a la integración de datos, el esfuerzo que toma, satisfacción general, y feedback. Para las preguntas a, b, c, y d se utilizó la misma escala Likert mencionada en el Anexo 5.

- ¿Cuán satisfecho estás con el resultado del proyecto respecto a la reducción de esfuerzo que necesitas para elaborar una demostración de AWS Supply Chain?
- ¿Cuán satisfecho estás con la industria seleccionada y el caso de uso para simular el funcionamiento de la herramienta?
- ¿Cuán satisfecho estás respecto a la demostración de las características del servicio AWS Supply Chain gracias a los datos generados?
- A modo general ¿Cuán satisfecho estás con el resultado del proyecto desarrollado
- ¿Encuentras que hay algo por mejorar en el proyecto?

Resultados disponibles en la imagen 28:

Encuestado	Pregunta			
	1	2	3	4
1	5	4	5	3
2	4	4	5	4
3	5	3	4	5
4	4	4	5	4
5	3	2	5	4
6	4	3	4	5
7	5	4	4	4
8	4	4	3	5
9	3	4	3	5
10	5	3	4	4
11	4	4	5	4
12	4	4	5	5
Ponderado	4,11	3,51	4,26	4,29
% Satisfacción	82%	70%	85%	86%

Imagen 28: resultados cualitativos de la solución 1

Referencias

- Anthropic. (2023, Julio 11). *Claude 2*. Anthropic. Retrieved Diciembre 03, 2023, from <https://www.anthropic.com/index/claude-2>
- AWS. (2022, Noviembre 29). *Connecting to a data source - AWS Supply Chain*. AWS Documentation. Retrieved Octubre 01, 2023, from <https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/data-connections.html>
- AWS. (2022, Noviembre 29). *Data entities supported in AWS Supply Chain - AWS Supply Chain*. AWS Documentation. Retrieved Octubre 01, 2023, from <https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/data-model.html>
- AWS - Amazon Bedrock. (2023, September 28). *Foundational Models - Amazon Bedrock - AWS*. Amazon AWS. Retrieved November 27, 2023, from <https://aws.amazon.com/bedrock/>
- AWS: Confidencial. (2022, Diciembre 10). *AWS Supply Chain - Demand planning dataset*.
- AWS Supply Chain. (2022). *Data entity columns required by Demand Planning - AWS Supply Chain*. AWS Documentation. Retrieved December 19, 2023, from https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/required_entities.html
- AWS Supply Chain. (2022). *Data entity columns required by Insights - AWS Supply Chain*. AWS Documentation. Retrieved Diciembre 01, 2023, from <https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/entities-insights.html>
- AWS Supply Chain. (2022, November 29). *Supply Chain Optimization — AWS Supply Chain — Amazon Web Services*. Amazon AWS. Retrieved October 07, 2023, from <https://aws.amazon.com/aws-supply-chain/>

- AWS Supply Chain - User guide. (2022, November 29). *Data entities supported in AWS Supply Chain - AWS Supply Chain*. AWS Documentation. Retrieved October 18, 2023, from <https://docs.aws.amazon.com/aws-supply-chain/latest/userguide/data-model.html>
- AWS - Well-Architected Framework. (2015, October 1). *AWS Well-Architected - Build secure, efficient cloud applications*. Amazon AWS. Retrieved October 10, 2023, from <https://aws.amazon.com/architecture/well-architected/>
- Hoory, L., & Bottorff, C. (2022, August 10). *Agile vs. Waterfall Methodology*. Forbes. Retrieved December 23, 2023, from <https://www.forbes.com/advisor/business/agile-vs-waterfall-methodology/>
- Investing.com & TradingEconomics.com. (n.d.). *Chile Government Bonds - Yields Curve*. World Government Bonds. Retrieved October 18, 2023, from <http://www.worldgovernmentbonds.com/country/chile/>
- Kaur, J. (2023, Noviembre 21). *Top Generative AI Use Cases and Applications*. XenonStack. Retrieved Octubre 12, 2023, from <https://www.xenonstack.com/blog/generative-ai-use-cases>
- Kvilhaug, S. (2023, May 24). *What Is the Capital Asset Pricing Model (CAPM)?* Investopedia. Retrieved October 18, 2023, from <https://www.investopedia.com/terms/c/capm.asp>
- OpenAI. (2022, November 30). *Introducing ChatGPT*. OpenAI. Retrieved November 27, 2023, from <https://openai.com/blog/chatgpt>
- Quinn, A. (2023, August 3). *13 Best Free Retail Datasets for Machine Learning*. Iguazio. Retrieved October 16, 2023, from <https://www.iguazio.com/blog/13-best-free-retail-datasets-for-machine-learning/>
- Richter, F. (2023, August 8). *Infographic: Amazon Maintains Lead in the Cloud Market*. Statista. Retrieved October 18, 2023, from <https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/>

- Teach For America. (2023, August 11). *The Promises and Perils of Generative AI in Education: TFA's Evolving Perspective*. Teach For America. Retrieved December 02, 2023, from <https://www.teachforamerica.org/one-day/ideas-and-solutions/the-promises-and-perils-of-generative-ai-in-education-tfas-evolving>
- Velasquez, V. (2023, May 24). *S&P 500 Average Return*. Investopedia. Retrieved October 18, 2023, from <https://www.investopedia.com/ask/answers/042415/what-average-annual-return-sp-500.asp>
- Waters, C. D. J. (2003). *Inventory Control and Management* (2nd ed.). Wiley.
- xenonstack. (2023, October 2). *Generative Adversarial Networks and Synthetic Patient Data*. XenonStack. Retrieved December 1, 2023, from <https://www.xenonstack.com/blog/gans-and-synthetic-patient-data>
- Yahoo Finance. (n.d.). *Amazon.com, Inc. (AMZN) Stock Price, News, Quote & History*. Yahoo Finance. Retrieved October 18, 2023, from <https://finance.yahoo.com/quote/AMZN/>

*“WORK HARD.
HAVE FUN.
MAKE HISTORY.”*

- Amazon