

## Construyendo una web API con la arquitectura n capas

### Building a web API with the n-layer architecture

María Alejandra Rosas Toro<sup>1</sup>  
Jaime Hesiquio Herrera<sup>2</sup>  
Leslie Suzette Velasco Salinas<sup>3</sup>  
Juan Manuel Martínez Chávez<sup>4</sup>

#### RESUMEN

En los procesos que involucran tecnologías digitales y la automatización, las empresas necesitan innovar para obtener sistemas informáticos eficientes y con mejores rendimientos; acorde con esto en la empresa BCD travel se desarrolló el proyecto “CONSTRUYENDO UNA WEB API CON LA ARQUITECTURA N CAPAS”, debido a que la herramienta de acceso a datos Data Service (DS) quedó descontinuada. El objetivo del proyecto era la creación de una Interfaz de Programación de Aplicaciones (API), capaz de mantener la comunicación y transmisión de información a cada uno de los procesos y sistemas que se ejecutan en esta empresa. La construcción de la Web API (*Herramienta de servicio de datos con Interfaz de programación de aplicaciones*) denominada DSA se basó en una arquitectura de N capas, utilizando la Tecnología Active Server Page (ASP .Net) y el lenguaje de programación C Sharp, aplicando el modelo de programación Object Relational Mapping (ORM) perteneciente al ADO.NET Entity Framework con la finalidad de mapear las estructuras de las Bases de Datos Relacionales y transformar las tablas para simplificar el trabajo de los desarrolladores. Los beneficios que ha tenido la DSA es que es una herramienta multitarea y permite ser utilizada en aplicaciones de escritorio, en sitios Web o en aplicaciones móviles, los tiempos de consulta de datos son más rápidos y se pueden realizar consultas en paralelo sin afectar el rendimiento de la API, un beneficio más de la DSA es que cualquier desarrollador puede darle mantenimiento, agregando o eliminando consultas a las tablas de la Base de Datos de acuerdo a las necesidades de procesamiento que se tengan.

**PALABRAS CLAVE:** Arquitectura de Software, Arquitectura N-Capas, *Servicio de Datos, Web Api*

**Fecha de recepción:** 13 de mayo, 2023.

**Fecha de aceptación:** 14 de julio, 2023.

<sup>1</sup> Docente de tiempo completo del Instituto Tecnológico de Cerro Azul. [maria.rt@cerroazul.tecnm.mx](mailto:maria.rt@cerroazul.tecnm.mx)

<sup>2</sup> Supervisor de analistas de sistemas de BCD Travel. [jaime.hesiquio@bcdtravel.com.mx](mailto:jaime.hesiquio@bcdtravel.com.mx)

<sup>3</sup> Desarrollador de Software Jr de BCD Travel. Exalumna del ITCA. [les.suztte@gmail.com](mailto:les.suztte@gmail.com)

<sup>4</sup> Analista de Sistemas en BCD Travel. Exalumno del ITCA. [jmanumc@outloo.com](mailto:jmanumc@outloo.com)

## ABSTRACT

In processes that involve digital technologies and automation, companies need to innovate to obtain efficient computer systems with better performance; Accordingly, in the company BCD travel, the project "BUILDING A WEB API WITH N LAYERS ARCHITECTURE" was developed, due to the fact that the Data Service (DS) data access tool was discontinued. The objective of the project was the creation of an Application Programming Interface (API), capable of maintaining the communication and transmission of information to each of the processes and systems that are executed in this company. The construction of the Web API (Data Service Tool with Application Programming Interface) called DSA was based on an N-layer architecture, using Active Server Page Technology (ASP.Net) and the C Sharp programming language, applying the Object Relational Mapping (ORM) programming model belonging to the ADO.NET Entity Framework in order to map the structures of Relational Databases and transform the tables to simplify the work of developers. The benefits that the DSA has had is that it is a multitasking tool and can be used in desktop applications, on Web sites or in mobile applications, data query times are faster and queries can be made in parallel without affecting performance From the API, one more benefit of the DSA is that any developer can maintain it, adding or removing queries to the Database tables according to the processing needs they have.

**KEYWORDS:** Data Service, N-Layer Architecture, Software Architecture, Web Api

## INTRODUCCIÓN

La empresa BCD Travel aunque tenía una herramienta de acceso a datos **DS** y que apoyaba en el desarrollo de aplicaciones de software Orientadas a Datos, esta ya no era funcional ya que dependía de otra herramienta externa que quedo obsoleta debido al avance de la tecnología, por lo tanto se desarrolló el proyecto denominado "CONSTRUYENDO UNA WEB API CON ARQUITECTURA N CAPAS", el objetivo era innovar con las nuevas tecnologías creando la "Data Service Api" (DSA) para sustituir al DS, esta nueva herramienta permite agilizar el trabajo de otros desarrolladores y consumidores de los diversas procesos que se manejan dentro de la empresa, optimizando el uso de los datos. La trascendencia de la Web API es que puede ayudar en gran medida en el futuro, dado que esta sirve de base para los siguientes proyectos de la empresa.

En este documento se describe el proceso que se llevó a cabo durante el desarrollo del proyecto, se inició por revisar y analizar el funcionamiento del **DS** (herramienta de acceso a datos con que se contaba), y procediendo a realizar el diseño y programación del software para el nuevo servicio de datos que lleva por nombre Data Service API (**DSA**), el cuál fue desarrollado en base a la plataforma Microsoft .NET Framework. Con esto se trata dar respuesta a los siguientes planteamientos:

- Se tiene la convicción de que la nueva herramienta DSA permitirá manejar todo tipo de información de manera eficiente para el ofrecimiento de servicios tanto para los empleados o desarrolladores de la empresa.
- Una nueva Web API permitirá mejorar el rendimiento y calidad de los servicios que la empresa ofrece.

## PLANTEAMIENTO DEL PROBLEMA

Dentro del departamento de Tecnología de la Información (**IT** por sus siglas en inglés) de la empresa BCD Travel, se encuentra el área de Desarrollo que se encarga de desarrollar y dar mantenimiento a todos los Sistemas Informáticos utilizados en la empresa. La problemática a la que se enfrentó esta área es que algunas de sus herramientas de acceso a datos han quedado descontinuados, tal es el caso de la DS que ya no cuenta con soporte técnico, ni mantenimiento por parte de la compañía que elaboraba las aplicaciones comerciales, por lo que era necesario crear una nueva herramienta la cual lleva por nombre Data Service API (DSA).

## Construyendo una web API con la arquitectura n capas

El DS depende del TELERIK ORM, que es otra herramienta externa que apoyaba en el desarrollo de aplicaciones de software orientadas a datos, y que ha sido descontinuada (Progress Telerik ORM, 2002) por lo que era imperante sustituirlo. Así mismo era necesario eliminar la arquitectura SOAP (Simple Object Access Protocol), ya que esta solo soporta el lenguaje XML (Extensible Markup Language) haciendo al DS más lento (IONOS, 2015), por lo tanto, había que reemplazarlo por un lenguaje más ligero y flexible como lo es el JSON (JavaScript Object Notation) que permite tener un código más limpio y legible (Bourhis, Reutter, & Vrgoc, 2020).

Dada la problemática, se propuso migrar el DS de la versión 1.0 a una nueva versión 2.0 con el nombre de Data Service Api (DSA) que contribuya a mejorar el servicio y estructura de la herramienta. El DSA sería un sistema multitareas para agilizar los tiempos de espera en la generación de reportes, para su desarrollo se sugirió el uso de la herramienta Entity Framework (EF) que es un ORM que permite el mapeo de Base de Datos (First Data Base) de tal manera que se permite mapear las Bases de Datos en forma de objetos y propiedades, y admitiendo actualizaciones y migraciones de esquemas (Singh, 2015). La arquitectura propuesta para la DSA fue la Representational State Transfer (REST) para permitir la comunicación entre el servidor y sus clientes, y que permite la escalabilidad, mejor rendimiento, facilidad de navegación, y flexibilidad (Fielding, 2000). Para el desarrollo de las aplicaciones Web que conforman el DSA se determinó usar la tecnología Active Server Page (ASP .Net), que permite crear servicios basados en HTTP a los que se puede acceder a aplicaciones en diferentes plataformas (Microsoft ASP.NET, 2020).

### JUSTIFICACIÓN.

El objetivo del proyecto “CONSTRUYENDO UNA WEB API CON LA ARQUITECTURA N CAPAS”, es la innovación de una nueva herramienta encargada de comunicar la información alojada en las Bases de Datos con distintos tipos de aplicaciones y sistemas a través de una red. Esta aplicación es muy importante para la empresa, ya que el servicio de acceso a la información DS quedó obsoleta y descontinuada, actualmente ya no se cuenta con soporte ni mantenimiento por parte de la compañía que se encargaba de la implementación de las aplicaciones comerciales. El proyecto permite reemplazar al DS por un servicio más eficiente, con menos costos de desarrollo mejorando el flujo de venta que la empresa maneja. El nuevo DSA cumple con los requerimientos y está basado en procesos que se adaptan a las necesidades actuales de la empresa, implementa nuevas tecnologías y usa la arquitectura de desarrollo de software de N capas; esto permite facilitar el mantenimiento del sistema, la integración de la Web Api a otros proyectos facilitando la escalabilidad y reduciendo los costos de desarrollo.

### METODOLOGÍA

Este proyecto corresponde a un estudio correlacional ya que por un lado se intenta describir la diferencia en cuanto a la eficiencia y seguridad que existe entre el DS y el DSA, así como la cuantificación de los beneficios relacionados con el rendimiento y la calidad de los servicios que presta la empresa. Con el estudio descriptivo, se desea analizar las características y condiciones que presenta el DSA con respecto al DS, conocer que tanto la nueva herramienta ahorra tiempo al acceder los datos, los beneficios que aporta a la empresa en cuanto a la calidad, seguridad, y confiabilidad; para los usuarios de la herramienta se desea determinar si la herramienta facilita el desempeño de su trabajo o no.

De lo planteado anteriormente, se puede decir que la investigación tiene un enfoque mixto integrando los métodos cualitativo y cuantitativo. Con el enfoque cualitativo se desea conocer la opinión de los desarrolladores al momento que experimenten el uso del DSA con respecto al DS, verificar si la nueva herramienta les facilita el trabajo que desempeñan, si se tiene una mejor estructura y código legible, y si la calidad de los resultados de los procesos mejora a los obtenidos con la DS. La recolección de datos en este enfoque es mediante la observación del 100% de los usuarios de la herramienta, con este se pretende conocer la realidad que impera en el uso de la DSA con respecto al DS. El enfoque cuantitativo, da a conocer el grado de eficiencia del uso de la nueva herramienta

## Construyendo una web API con la arquitectura n capas

DSA con respecto a la herramienta DS, el método de recolección de datos para medir las variables es una encuesta estructurada compuesta de 10 preguntas de opción múltiple aplicada a todos los usuarios de la herramienta y que arroja los datos cuantitativos requeridos para aceptar o rechazar las hipótesis planteadas.

### HIPÓTESIS

Las hipótesis que se plantean al implementar el proyecto son:

**H<sub>11</sub>**: La herramienta Data Service Api (DSA) permitirá manejar todo tipo de información, de manera eficiente para ofrecimiento de servicios ya sea para los desarrolladores y/o clientes.

**H<sub>A1</sub>**: La herramienta Data Service Api no permitirá manejar todo tipo de información de manera eficiente para ofrecimiento de servicios ya sea para los desarrolladores y/o clientes.

**H<sub>12</sub>**: La creación e implementación de la Web API como alternativa de solución, permitirá mejorar el rendimiento y calidad de servicios que la empresa ofrece.

**H<sub>A2</sub>**: La creación e implementación del Web API como alternativa de solución, no permitirá mejorar el rendimiento y calidad de servicios que la empresa ofrece.

Los pasos que se siguieron en el desarrollo del proyecto se describen a continuación:

**Planificación.** En el desarrollo del DSA primeramente se establecieron las prioridades del proyecto y se estimó el orden de las actividades. En esta etapa se revisó la DS y se analizó la viabilidad de crear la nueva herramienta que diera solución a los problemas del sistema actual, se definieron los objetivos del nuevo sistema y se identificaron los recursos necesarios para el desarrollo de la misma.

**Análisis de requerimientos.** Al realizar el análisis se detectó la necesidad de identificar los paquetes de datos contenidos en las Bases de Datos, las funcionalidades que estos tenían y reestructurarlos para que fueran más funcionales y efectivos en la solución de una gran cantidad de problemas. Se requirió el establecimiento de los límites de la DSA identificando los agentes externos con los que interacciona y remplazar la DS, predeterminando que la DSA debía ser una mejor herramienta de servicio de datos adaptable a los distintos procesos informáticos de la empresa.

**Diseño.** En esta etapa se analizaron diversas alternativas de implementación para el DSA, eligiendo la estructura general de consulta de datos con una Arquitectura de N capas, ya que la herramienta debe manejar todo tipo de información que utiliza la empresa, ya sea de clientes, de proveedores o empleados (Richards, 2015). El DSA usa 5 capas: capa de Usuario (persona física que interactúa con el sistema), capa de Servicios (Sistema interfaz que permite al usuario realizar peticiones a la Web Service API), capa de Presentación (atiende las peticiones que el usuario realiza como es agregar, eliminar, o consultar datos), capa de Lógica de Negocio (valida que los datos sean correctos) y capa de Datos (realiza las consultas a la base de datos), ver Figura 1.

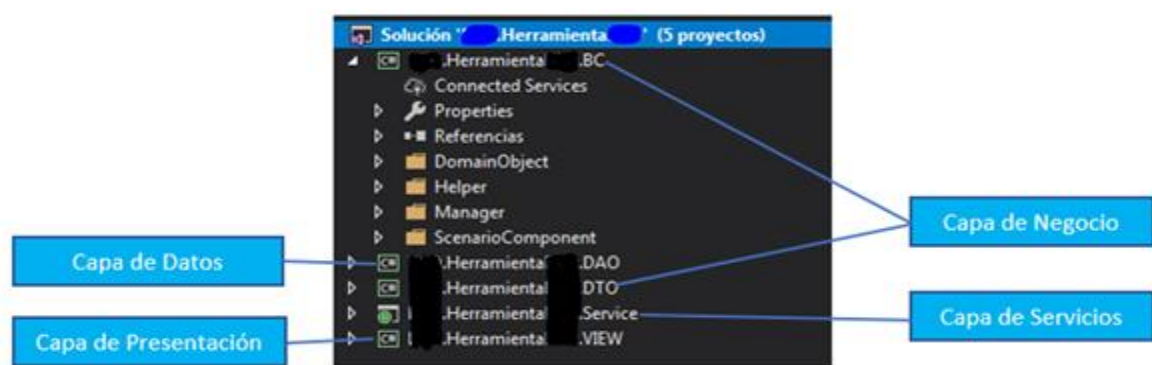


Figura 1. Arquitectura de N-Capas del proyecto.

Construyendo una web API con la arquitectura n capas

La Figura 2, muestra el proceso para obtención de datos en la arquitectura de N-Capas, el proceso inicia en la capa de presentación donde el usuario indica cual es la información que necesita recuperar, luego viaja a la capa de datos donde realiza la consulta, posteriormente en la capa de lógica de negocio se validan los datos y finalmente regresa hasta la capa de presentación donde muestra al usuario el resultado de los datos obtenidos. El proceso de actualizar registros también se inicia en la capa de presentación, donde el usuario indica el registro y los datos que se desean actualizar, de ahí se viaja a la capa de datos en donde se realiza la búsqueda del registro y se reemplazan los datos, posteriormente regresa a la capa de presentación donde se muestra al usuario el mensaje de “Registro actualizado correctamente”.

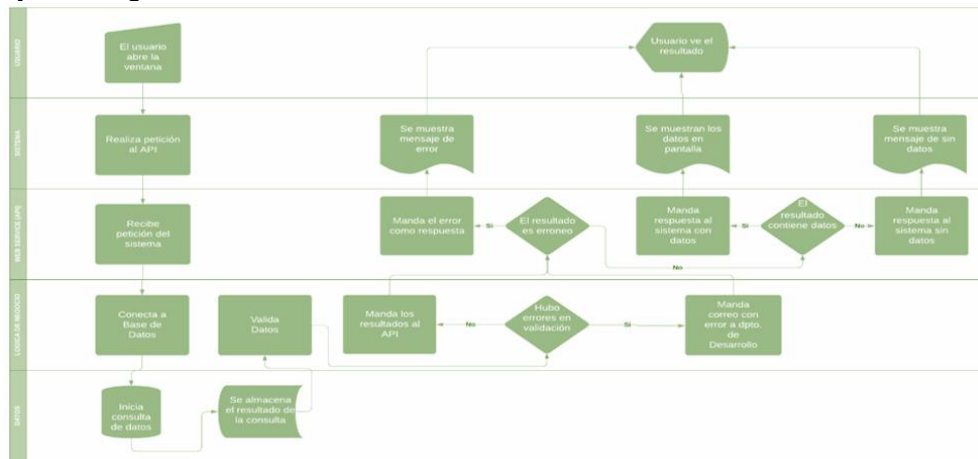


Figura 2. Diagrama de Flujo del proceso de obtención de datos.

La eliminación de registros se inicia en la capa de presentación, donde el usuario indica el registro que se debe eliminar, de ahí viaja a la capa de datos realizando la búsqueda y eliminación del registro de la base de datos, posteriormente se regresa a la capa de presentación donde se indica al usuario que se realizó la eliminación de forma correcta. El DSA se usa directamente con las tablas de las Bases de Datos de la empresa, ya que estas son robustas y cuentan con el soporte necesario. La Base de Datos “Producción” contiene tablas con datos reales que son aprovechados por los usuarios finales, la Base de Datos “Calidad” contiene tablas que son utilizadas por los tester de la empresa para validar los criterios de aceptación del software implementado por los desarrolladores, y la Base de Datos “Desarrollador” contiene tablas usadas por los desarrolladores. La Figura 3, muestra el diagrama de conexión a la Base de Datos.

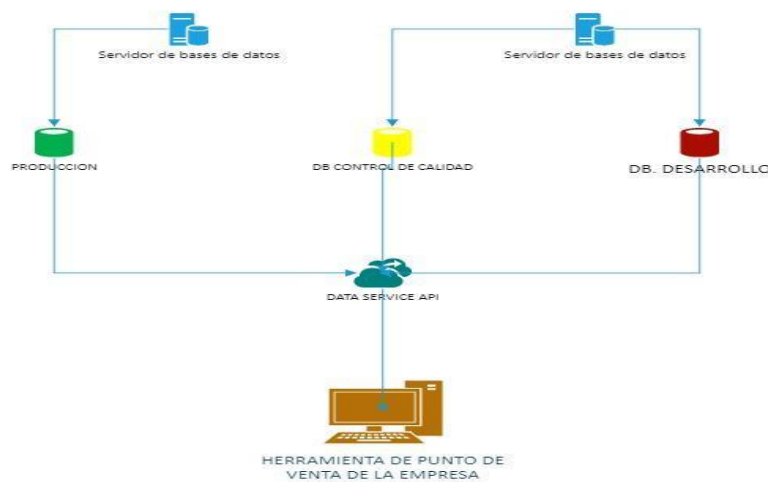


Figura 3 Conexión a Base de Datos



## Construyendo una web API con la arquitectura n capas

**Codificación.** El DSA fue construido y montado sobre el Framework *.NET* (Microsoft *.NET*, 2020) de la cual la empresa cuenta con una licencia. Para la codificación se usó el entorno de desarrollo de software Visual Studio de Microsoft por la facilidad de manipulación y las herramientas que este tiene para la edición de los programas. El lenguaje de programación para cada uno de los programas de los diversos módulos fue *C Sharp* ya que este permite a los desarrolladores escribir programas orientados a componentes con seguridad de tipos y que pueden ejecutarse en *.NET*, este lenguaje permite la creación de aplicaciones en consola, con interfaz gráfica de usuario (GUI), para dispositivos móviles, servicios de API y páginas Web con la tecnología ASP.NET.

Dentro de los módulos de la DSA la capa de acceso a datos o *Data Access Object (DAO)*, contiene la lógica para la conexión y administración de la base de datos utilizando la Entity Framework (Herramienta de acceso a datos) con el modo Data Base First que permite mapear tablas y procedimientos almacenados a través del componente ADO.NET (Vaikunth & Aithal, 2017). La capa de negocios o Business Component (BC) contiene la lógica de negocios, conectando las capas DAO y Service, iniciando la interacción con la base de datos de acuerdo a una petición que llega a través del Service. Esta capa contiene tres subcapas Domain Object (DO) que conecta la DAO con la subcapa *Scenary Component (SC)* haciendo los métodos más sencillos, el SC contiene la lógica principal de los métodos que consultan a la base de datos y el Manager funciona como método de acceso de la capa Service a la subcapa SC.

El Service es la última capa del proyecto, funciona como punto de acceso al programa, ya que esta es la encargada de exponer los métodos programados en la capa BC a través de la red utilizando el protocolo HTTP, esta capa es lo que convierte todo el programa en una API. La capa Controllers contiene varias clases con distintas funcionalidades para la API, estas son accesibles desde la red. Una característica del controlador son los parámetros que necesitan los métodos de la subcapa Manager para funcionar. Response es un estándar que se programó, tomando en cuenta el JSONAPI para que la API siempre devuelva la misma estructura de respuesta en formato JSON, se tomó la estructura base que llevan las respuestas exitosas y las respuestas de error. Como resultado se obtuvo una herramienta con código limpio y legible para que cualquier desarrollador con conocimiento básico en el lenguaje de programación C Sharp y el Framework *.NET*, pueda dar soporte y mantenimiento a la Web API sin mucho esfuerzo y tiempo invertido.

### Pruebas y evaluación.

El objetivo de las Pruebas fue detectar los errores cometidos durante las actividades anteriores y modificarlos antes de que la DSA llegara al usuario final. Con ayuda de librerías que Microsoft ofrece se decidió hacer uso de las Pruebas Unitarias, estas consisten en aislar una parte del código y comprobar que funciona de la manera correcta. Las pruebas unitarias se desarrollaron antes de generar un código, esto permitió ir solucionando los errores conforme se avanzaba, y así se evitó retraso a la hora de sacar el proyecto a producción.

Las pruebas unitarias son capaces de descomponer los módulos de la DSA en comportamientos comprobables discretos, que se pueden probar como unidades individuales. El explorador de pruebas de Visual Studio proporciona una forma flexible y eficaz de ejecutar las pruebas unitarias y ver los resultados. Al definir y ejecutar las pruebas unitarias o test, se garantiza la cobertura del código, se detectan errores y posibles fallos antes de que lo hagan los clientes. La Figura 4 muestra la estructura de una prueba unitaria.

## Construyendo una web API con la arquitectura n capas

```
using Microsoft.VisualStudio.TestTools.UnitTesting;
using System.Net.Http;

namespace [redacted].DataServiceApi.Test.Release
{
    [TestClass]
    0 referencias
    public class EjemploTest : Test
    {
        0 referencias
        public void ejemplo_ShouldOk()
        {
            HttpResponseMessage response = HmacClient.GetAsync("api/example").Result;
            Assert.IsTrue(response.IsSuccessStatusCode);
        }
    }
}
```

Figura 4 Estructura de una prueba unitaria

### Mantenimiento.

El mantenimiento consiste en eliminar los defectos que se detectaron durante la vida útil de la DSA y adecuar a las nuevas necesidades. Actualmente la API ya se encuentra publicada y en prueba, durante el transcurso de las pruebas surgen errores los cuales son detectados gracias a un método que permite enviar un correo al equipo de desarrolladores anexando el problema que suceda y en donde se suscitó. Para corregir un defecto se revisa con que Status Code responde el DSA, y de acuerdo al status se le da el seguimiento. Algunos status y procedimiento a seguir son:

- **400 petición Incorrecta.** El servidor no puede o no procesará la solicitud debido a un aparente error del cliente; en este caso es necesario ver que datos se envían y si se cumplen las validaciones que se implementaron en ese método.
- **401 no autorizado.** Se requiere autenticación y ha fallado o aún no se ha proporcionado. El usuario no tiene credenciales de autenticación válidas para el recurso de destino.
- **404 no encontrado.** El recurso solicitado no se pudo encontrar, pero puede estar disponible en el futuro, para solucionar este error el encargado de la administración de base de datos debe agregar dicha tabla o ingresar nuevos registros a la tabla.
- **500 error interno de servidor.** Un mensaje de error genérico se da cuando se encontró una condición inesperada, la única forma de solucionarlo es reportar que el servidor no se encuentra funcionando y esperar a que éste sea solucionado por los encargados del soporte operativo.

Cuando se está preparando un programa para un entorno de producción es necesario tener un Log donde se reportan los eventos y errores, la DSA genera los archivos Logs los cuales son atendidos de manera rápida para darle mantenimiento a la aplicación antes de pasar a producción.

### DISCUSIÓN DE RESULTADOS.

El análisis de correlación entre las herramientas Data Service y Data Service Api se llevó a cabo con el equipo de desarrolladores de la empresa, a cada desarrollador se les asignó tareas aplicables a los sistemas que cada uno maneja, solo que estas debían ser realizadas primero implementando el DS y posteriormente el DSA. Para comparar los resultados se creó una tabla de registro de productividad del DS y otra para el DSA, de acuerdo a los datos registrados se pudo observar que el DSA es una herramienta innovadora para la empresa, fácil de usar y de fácil mantenimiento, ya que a los desarrolladores no les costó trabajo adaptarla a sus sistemas, en comparación con el DS en donde solo 2 personas les costó un poco de trabajo implementarla. También se pudo observar que la DSA mantiene los datos más seguros al mapear las tablas de las Bases de Datos y transformarlas a la estructura necesaria para cada sistema y tratarlos como objetos, acción que el DS no puede llevar a cabo, esto hace a la DSA una herramienta más eficiente y segura, la cual tiene

**Construyendo una web API con la arquitectura n capas**

mejores rendimientos permitiendo que la generación de los reportes sea de mejor calidad, la herramienta es escalable y reduce los costos de desarrollo.

Para determinar si las hipótesis planteadas son aceptadas o no, se aplicó una encuesta estructurada compuesta por 10 preguntas de opción múltiple a los 10 desarrolladores de la empresa que son los que conocen el funcionamiento del DS. Con los datos recabados se obtuvieron las tablas y gráficos de frecuencias estadísticas y porcentajes de utilidad de cada una de las preguntas. Para calcular el porcentaje de utilidad se usó la siguiente ecuación.

$$\%Utilidad = \frac{Frecuencia\ de\ utilidad}{Total\ Encuestados}$$

De acuerdo a la pregunta 1 ¿Es útil el uso del Data Service Api para usted?, a 8 de los desarrolladores les parece muy útil el DSA en comparación con el DS, esto representa el 80% de utilidad para la empresa en comparación al DS lo que permite reducir los costos de desarrollo, sólo 2 de los desarrolladores no pudieron determinar si le es útil la herramienta o no, (Ver Tabla 1).

**Tabla 1. Frecuencia de Utilidad del DSA.**

¿Es útil el uso de Data Service Api para usted?	Frecuencia	% de Utilidad
<b>Sí, mucho</b>	8	80%
<b>No sé</b>	2	20%
<b>No</b>	0	0%
<b>Era mejor DS</b>	0	0%

En cuanto al aspecto profesional de la interfaz gráfica de la DSA, se tiene la pregunta 2 ¿Cómo le parece el aspecto profesional de la interfaz de la DSA?, el 20% de los desarrolladores consideran que es muy profesional, el 50% expresaron que el aspecto es profesional y solo un 30% la consideran no profesional (ver Tabla 2), esto indica que se puede seguir trabajando en la herramienta para darle una interfaz más amigable. De acuerdo a los resultados se puede observar que hay una diferencia muy grande entre la interfaz gráfica de la DSA y la DS, siendo la DSA más fácil de usar e intuitiva.

**Tabla 2. Frecuencia de opiniones en cuanto al aspecto profesional de la DSA**

Como le parece el aspecto profesional de la interfaz de la DSA	Frecuencia	%
<b>Muy profesional</b>	2	20%
<b>Profesional</b>	5	50%
<b>No muy profesional</b>	3	30%
<b>No profesional</b>	0	0%

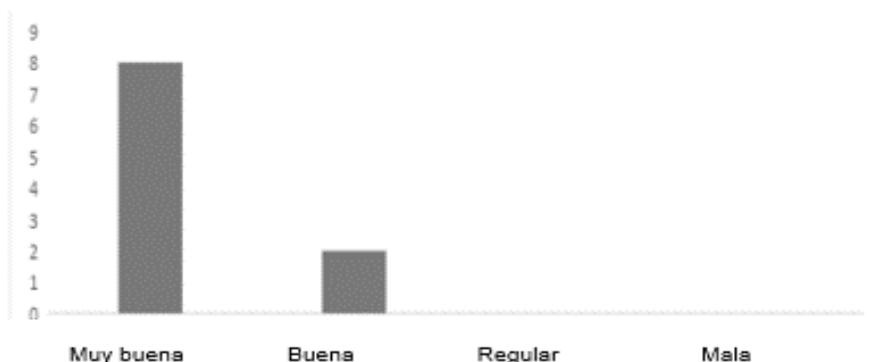
En cuanto a la calidad de los resultados obtenidos en los procesos que los desarrolladores ejecutaron aplicando la herramienta DS y la DSA, se tiene la Pregunta 3 ¿Cuál es la calidad del trabajo del Data Service Api en comparación del Data Service?, el 80% de los desarrolladores consideran que la calidad de los resultados obtenidos con la DSA es muy buena, el 20 % consideran que la calidad es buena, todos coinciden que hay mejor calidad en los resultados obtenidos con la DSA como se muestra en la Figura 5. Esto prueba que hay en avance significativo y muy seguro en la solución de problemas con el uso de la DSA en comparación con la DS.





## Construyendo una web API con la arquitectura n capas

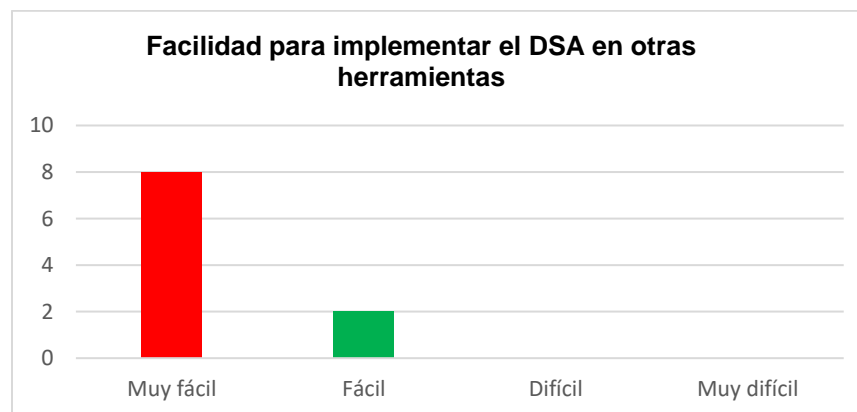
¿Cuál es la calidad de trabajo de Data Service Api en la comparación de Data Service?



**Figura 5 Calidad de los resultados aplicando la Data Service Api**

Para comparar cuál de las dos herramientas es más fácil de usar de acuerdo a su interfaz gráfica, se creó la pregunta 4 ¿La Data Service Api es más sencilla de usar que la Data Service? De acuerdo a la información recopilada se puede observar que el 80% de los desarrolladores consideran que la DSA es más fácil de usar en comparación con la DS y el 20% consideraron que, si les fue un poco más sencillo, pero ninguno expresó que se le haya dificultado.

Para comparar la eficiencia de implementar la DSA en otras herramientas y procesos que se llevan a cabo en la empresa, se realizó la pregunta 5 ¿El Data Service Api es más sencillo de implementar en otras herramientas? El 80% de los desarrolladores consideran que la DSA es mucho más sencilla de implementar en otras herramientas en comparación con la DS, 2 desarrolladores consideran que es un poco más sencillo, y a ningún desarrollador se le hizo difícil de implementar (Ver Figura 6). Con las actividades que realizaron los desarrolladores, todos de manera fácil pudieron adaptar el DSA a los procesos que cada uno tienen asignados.



**Figura 6 Facilidad de uso del DSA**

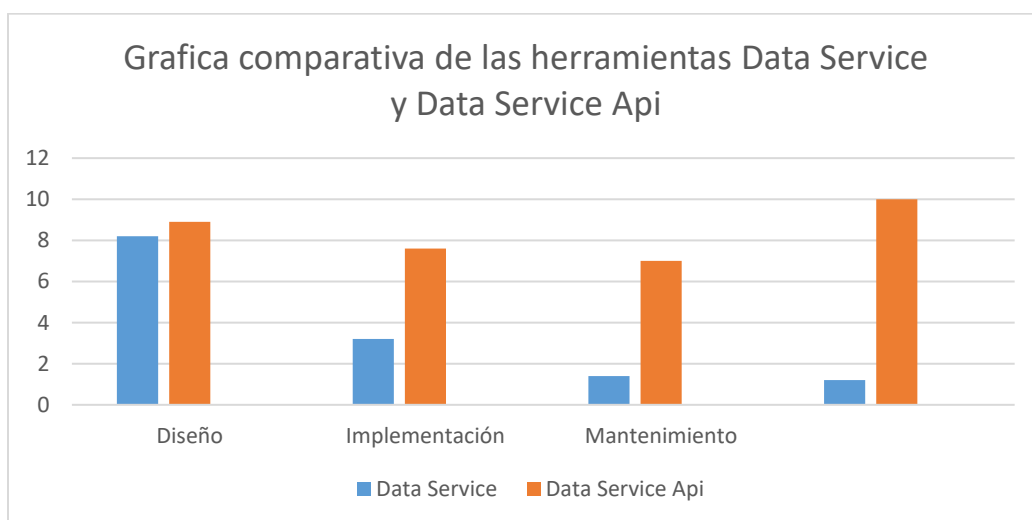
La pregunta 6 ¿Es más fácil realizar pruebas en el Data Service Api que en el DS?, va orientada a la serie de pruebas que deben llevarse a cabo cuando algún proceso necesita de nueva información que se tiene alojada en las Bases de Datos. Al 80% de los usuarios se les hizo más fácil y al 20% se les hizo fácil realizar las pruebas con la DSA en comparación con el DS, a ningún desarrollador se le hizo difícil realizar las pruebas. En cuanto a la pregunta 7 ¿En la Data Service Api se producen menos errores que en Data Service?, 9 de los 10 desarrolladores consideran que el DSA produce menos errores en los procesos en comparación con la DS, sólo un desarrollador percibió que se

### Construyendo una web API con la arquitectura n capas

tienen los mismos resultados con ambas herramientas. Esto representa un 90% de eficiencia y seguridad de la información de la nueva herramienta con respecto a la herramienta anterior.

Para determinar cual herramienta ya sea DSA o DS usaria cada uno de los desarrolladores, se planteó la pregunta 8 ¿Si tuviera que elegir entre el Data Service Api y el Data Service para implementarlo en sus procesos, con cual se quedaría?, el 80% de los usuarios eligirian la nueva herramienta para el desarrollo de su trabajo, ya que la han probado y han experimentado que esta les facilita su labor. Con la pregunta 9 ¿El Data Service Api cumple con todas las expectativas de una nueva herramienta? Aunque los usuarios han experimentado los beneficios que han logrado con el DSA, sólo el 20% de ellos consideran que la DSA cumple con todas las expectativas, y otro 50% consideran que aún se pueden tener más mejoras en la herramienta.

La preunta 10 ¿Del Data Service y el Data Service Api, cuál le resulta más sencillo para ayudar a su mantenimiento?, esta pregunta va orientada a la facilidad que tendrán los desarrolladores para realizar los cambios pertinentes a la herramienta, para que esta se adapte a la necesidad que cada proceso tenga en cuanto a información almacenada en las Bases de Datos. El 100% de los desarrolladores consideraron que no tienen ningún problema en modificar el código de la nueva herramienta y adatarlo a sus necesidades. Con los resultados de la encuesta se puede observar que el DSA ha cumplido de manera exitosa con los objetivos propuestos y que aún existen áreas de oportunidad para mejorarla. La Figura 7 muestra la comparación de las dos herramientas en cuanto a su diseño, implementación y mantenimiento.



**Figura 9 Gráfica comparativa de las herramientas DS y DSA**

Como se observa en la gráfica la DSA es más eficiente que DS. Basados en las necesidades de la empresa se innovo en una nueva herramienta que permite manejar todo tipo de información de manera eficiente y segura, para ofrecer servicios a los empleados y/o clientes, con esto se puede determinar que la hipótesis  $H_{11}$  es Aceptada. En cuanto a la hipótesis  $H_{12}$ , la creación e implementación del web API como alternativa de solución, permitirá mejorar el rendimiento y calidad de servicios que la empresa ofrece, respaldados en los resultados obtenidos en la encuesta en las preguntas 1, 3, 4, 5, 6, 8 se obtuvo un 80% de eficiencia, en la pregunta 7 el 70% de los usuarios determinaron que el API produce menos errores y en la pregunta 10 el 100% de los usuarios prefieren la nueva herramienta, por lo que se concluye que esta hipótesis es Aceptada.

## CONCLUSIÓN

El proyecto se enfocó en el aprovechamiento de las nuevas tecnologías Web API REST, con la intención de proporcionar a las herramientas que maneja la empresa la facilidad de utilizar la lógica de negocios implementada en la aplicación, incrementando la reutilización de métodos, y permitiendo que la información de la herramienta web esté disponible para que otros desarrolladores puedan utilizarla en sus sistemas. Se desarrolló una plataforma independiente para el registro de la productividad, y no dependiente de otros proyectos o herramientas de la empresa. Al realizar la comparación de los resultados, se detectó un incremento en la cantidad de datos aprobados, lo que se traduce a un mayor apoyo a futuros desarrolladores, siendo esta nueva herramienta una innovación para los procesos que se desarrollan en la empresa y que han servido de preambulo para que la DSA sea aplicada en los procesos que se llevan a cabo en las sucursales que se encuentran en Sudamérica. La DSA es una herramienta que cualquier desarrollador puede darle mantenimiento, puede insertar, modificar y eliminar consultas a las tablas de las Bases de Datos, las consultas a datos se llevan en menor tiempo y se pueden hacer consultas en paralelo.

Considerando los resultados que se han obtenido con la aplicación de la DSA, se llegó a la conclusión que la Web API es de gran ayuda para nuevos sistemas creados por los desarrolladores ya que esta reduce el tiempo de mapeo y configuración de la información, proporciona seguridad a las Bases de Datos, reduce los gastos de mantenimiento, y optimiza el espacio de almacenamiento de manera eficiente y funcional. Aunque las hipótesis planteadas han sido aceptadas, aún falta por mejorar la herramienta, por lo que se concluye que la Data Service Api es un área de oportunidad para seguir explorando y realizar mejoras que aún puedan incrementar la calidad de los resultados de dicha herramienta

## REFERENCIAS

- Bourhis, P., Reutter, J. L., & Vrgoc, D. (2020). *JSON: Data model and query languages* (Vol. 89). Information Systems. doi:doi.org/10.1016/j.is.2019.101478.
- Ecma International. (Diciembre de 1999). *"ECMAScript Language Specification", Standard ECMA-262, Third Edition*. Obtenido de ecma-international.org: <http://www.ecma-international.org/publications/files/ECMA-ST-ARCH/ECMA-262,%203rd%20edition,%20December%201999.pdf>
- Fielding, R. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Obtenido de ics.uci.edu: [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm)
- IONOS. (20 de Abril de 2015). *SOAP: explicación del protocolo de red*. Obtenido de Digital Guide IONOS by 1&1: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/soap-simple-object-access-protocol/>
- Microsoft .NET. (1 de Marzo de 2020). *What is .NET?* Obtenido de .NET: <https://dotnet.microsoft.com/learn/dotnet/what-is-dotnet>
- Microsoft ASP.NET. (9 de Marzo de 2020). *What is ASP.NET?* Obtenido de .NET: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>
- Progress Telerik ORM. (2002). *Telerik OpenAccess Classic*. Obtenido de Telerik Documentation: <https://docs.telerik.com/help/openaccess-classic/getting-started-with-openaccess-overview.html>

Richards, M. (2015). *Software Architecture Patterns*. ÓReilly Media, Inc.

Singh, R. R. (2015). *Mastering Entity Framework*. (P. E. Distilled, Ed.) Packt Publishing.

Vaikunth, P. T., & Aithal, P. S. (2017). *Disconnected Data Access Architecture using ADO.NET Framework*. International Journal of Applied Engineering and Management.  
doi:<http://dx.doi.org/10.5281/zenodo.87357>