



TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN SISTEMAS DE INFORMACIÓN



Análisis y estudio de la aplicación de Buenas prácticas en las funciones de Testing en el ámbito de los Sistemas de Inteligencia Empresarial (BI)

Estudiante: Cristina Calvo Rodríguez

Dirección: María Martínez Pérez

Ángel Gómez García

A Coruña, Septiembre de 2023.

A mi familia

Agradecimientos

A mis padres, por haberme dado la oportunidad de estudiar y por el apoyo constante recibido a lo largo de los años. Estoy muy agradecida por todo lo que me habéis y seguís aportando y ayudando día a día.

A mis jefes y compañeros de trabajo, en especial a Miguel, por ser una fuente inagotable de apoyo, consejos, ánimos e insistencia.

A mis directores, por haberme guiado y aconsejado durante todo el proyecto.

A todos los profesores que me han ayudado a llegar hasta este momento.

Resumen

Los Sistemas de Business Intelligence permiten a las empresas consolidar los datos que poseen en sus diferentes áreas funcionales y transformarlos en información de alto valor para la toma de decisiones de la compañía.

Sin embargo, estos sistemas deben ser probados en cada una de sus etapas y módulos para garantizar su buen funcionamiento, en especial cuando los sistemas contienen grandes volúmenes de datos.

Durante la extracción de los datos, estos deben ser de calidad para que, en el momento de transformarlos, se pueda seguir garantizando su fiabilidad. Las herramientas de Testing de BI permiten conocer el estado del funcionamiento de cada módulo. Algunas de estas herramientas solo prueban una parte del sistema de BI, mientras que existen otras que pueden probar todos los módulos, es el caso de la herramienta NBi, la cual fue elegida para realizar las pruebas de testing en el proyecto.

Este proyecto nace ante la necesidad detectada en un entorno real de agilizar las pruebas de testing en un desarrollo crítico, donde actualmente todas las pruebas se realizan de forma manual.

Con NBi se pueden probar los datos desde que son recolectados, procesados y transformados en el módulo de ETL, además de probar los informes que se generarán y los cuadros de mando. Del mismo modo, se pueden probar los datos directamente en la base de datos y así asegurarse de que estos son de calidad y fiables y podrán aportar información que sea de alto valor para la toma de decisiones.

Dicho esto, se puede confirmar que las herramientas de testing son cruciales en un proyecto de Business Intelligence.

Abstract

Business Intelligence Systems allow companies to consolidate the data they have in their different functional areas and transform it into high-value information for the company's decision making.

However, these systems must be tested in each of their stages and modules to ensure their proper functioning, especially when the systems contain large volumes of data.

During data extraction, the data must be of high quality so that when it is transformed, its reliability can still be guaranteed. BI testing tools allow to know the performance status of each module. Some of these tools only test a part of the BI system, while others can test all the modules, as is the case of the NBI tool, which was chosen to perform the tests in the project.

This project was born from the need detected in a real environment to streamline testing in a critical development, where currently all tests are performed manually.

With NBI you can test the data from the moment it is collected, processed and transformed in the ETL module, as well as test the reports that will be generated and the dashboards. Similarly, data can be tested directly in the database to ensure that it is of high quality, reliable and will be able to provide information that is of high value for decision making.

With all this said, it can be confirmed that testing tools are crucial in a Business Intelligence project.

Palabras clave:

- Inteligencia de Negocio
- Herramientas de Testing BI
- Testing
- NBI.io

Keywords:

- Business Intelligence
- BI Testing Tools
- Testing
- NBI.io

Índice general

1	Introducción	1
1.1	Objetivos	3
1.1.1	Objetivo general	3
1.1.2	Objetivos detallados	3
1.2	Organización de la memoria	4
2	Metodología	6
2.1	Metodología SCRUM	6
2.1.1	Roles	7
2.1.2	Actividades o eventos	8
2.1.3	Herramientas o artefactos	8
2.2	Adaptación de la metodología Scrum al proyecto	9
2.3	Gestión del proyecto	9
2.3.1	Planificación y estimación del proyecto	10
2.3.2	Estimación	10
2.3.3	Costes	12
2.3.4	Gestión de riesgos	13
3	Inteligencia de Negocio	14
3.1	Definición de Inteligencia de Negocio o Business Intelligence	14
3.2	Áreas de aplicación	15
3.3	Características	17
3.4	Componentes	18
3.4.1	Fuentes de información	19
3.4.2	ETL	19
3.4.3	Datawarehouse y/o Datamart	20
3.4.4	Servidor OLAP	20
3.4.5	Herramientas de visualización	20

3.5	Beneficios de la Inteligencia de Negocio	21
4	BI en las empresas	24
4.1	Implementación de BI en las empresas	24
4.2	Modelo de Madurez de Business Intelligence	26
4.3	Minería de datos (Data Mining)	27
4.4	Procesos de minería de datos	28
5	Testing en BI	31
5.1	Definición de Testing	31
5.2	Tipos de testing en Business Intelligence	32
5.3	Tipos de licencias para software de BI	34
5.4	Herramientas de testing existentes en el mercado para BI	34
5.5	Ventajas de las pruebas de BI	36
6	Herramientas de Testing en BI	38
6.1	TMF: Herramienta para Ruby de pruebas mínimas	38
6.1.1	Uso de TMF	38
6.2	Motio PI para Cognos de IBM	39
6.2.1	Bondades de Motio PI	39
6.2.2	Algunos test de Motio PI	39
6.3	RightData	40
6.3.1	Beneficios de las pruebas de ETL con RightData	40
6.4	QuerySurge	41
6.4.1	Ventajas que ofrece QuerySurge	41
6.5	Datagaps BI Validator	42
6.5.1	Beneficios con Datagaps BI Validator	43
6.6	NBi	43
6.6.1	Funcionamiento	44
6.6.2	Coste de NBi	44
6.6.3	Ventajas del uso de NBi	45
7	Caso de estudio: Aplicación NBi	46
7.1	Contextualización	46
7.2	Desarrollo y ejecución de las pruebas	48
7.2.1	Definición de las pruebas	48
7.2.2	Planificación y buenas prácticas	49
7.2.3	Creación y ejecución de las pruebas	50
7.3	Comparativa y conclusiones	57

7.3.1	Validación manual	57
7.3.2	Conclusiones sobre la validación	59
8	Conclusiones y trabajos futuros	62
8.1	Conclusiones	63
8.2	Trabajos futuros	64
A	Guía de instalación	66
A.1	Instalación manual	66
A.2	Instalación mediante Visual Studio	69
B	Test de prueba básicos con NBi	72
	Lista de acrónimos	82
	Glosario	84
	Bibliografía	85

Índice de figuras

2.1	Metodología SCRUM	7
2.2	Diagrama de Gantt. Planificación del proyecto	11
3.1	Componentes de Business Intelligence	19
4.1	Metodología CRISP-DM.	30
7.1	Test de prueba con AdventureWorks2022	49
7.2	Caso de estudio. Test erróneo	56
7.3	Caso de estudio. Explicación error	56
7.4	Caso de estudio. Ejecución correcta del test	57
7.5	Validación. Comprobación con tablas dinámicas	58
7.6	Validación. Comparación de las exportaciones Excel	58
A.1	Instalación NBi. Organización directorio base	67
A.2	Instalación NBi. Creación del proyecto en Visual Studio 2022	70
A.3	Instalación NBi. Consola del administrador de paquetes	71

Índice de tablas

2.1	Coste materiales	12
2.2	Coste recursos humanos	13
2.3	Coste total	13
7.1	Tiempos de validación.	59

Listings

6.1	Ejecución de test en TMF	38
7.1	Plantilla de prueba NBI	51
7.2	Definición de la cadena dentro del test	51
7.3	Definición de la cadena por defecto	52
7.4	Definición de la cadena por referencia	52
7.5	Definición de las cadenas en fichero de configuración (1/2)	52
7.6	Definición de las cadenas en fichero de configuración (2/2)	53
7.7	Definición del conjunto de resultados en un archivo externo	53
7.8	Definición del conjunto de resultados en el test	53
7.9	Definición del conjunto de resultados en un archivo XML	54
7.10	Definición de la comparación directa entre queries	54
7.11	Caso de estudio. Test con NBI	55
7.12	Caso de estudio. Réplica del test en NBI para cada empresa	60
A.1	Configuración del archivo .config	68
A.2	Configuración del archivo .nunit	68
A.3	Configuración del archivo .nbits	68
A.4	Comando de instalación de NBI en Visual Studio	71
B.1	Test de prueba básico con NBI	72

Introducción

La velocidad con la que cambian los mercados actualmente es vertiginosa. Las empresas cada día generan grandes cantidades de datos en sus sistemas que deben ser procesados y analizados lo más rápido posible. Sin embargo, en muchos casos no disponen de los mecanismos que les permitan realizar esta tarea de extracción de conocimiento para la ayuda a la toma de decisiones.

A pesar de que cada organización debe adaptarse a los cambios que surjan en el tiempo y a las nuevas tecnologías, con términos como la globalización, los cambios son constantes y muy rápidos, por lo que las empresas deben buscar nuevas oportunidades para no desaparecer por no poder lograr una adaptación eficiente.

En las compañías pueden existir múltiples sistemas de información, los cuales pueden ser independientes entre sí o estar interconectados [1]. Las empresas, a veces, poseen sistemas de información que no están debidamente implementados y no cubren las necesidades de la compañía, mientras que otras veces, el problema no está en cómo se encuentra implementado el sistema de información o el tipo de sistema, sino que no se le hacen las preguntas correctas o no se utilizan las herramientas adecuadas.

Los sistemas transaccionales son definidos como *"un conjunto de componentes interrelacionados que permiten capturar, procesar, almacenar y distribuir la información para apoyar la toma de decisiones y el control en una institución"* [2]. Estos están enfocados en la toma de decisiones, donde se tratará la información con la finalidad de ayudar y apoyar al usuario en la toma de decisiones [1].

Aunque los sistemas transaccionales nos pueden ayudar en algunos casos a tomar una decisión, no siempre es igual. Actualmente, en el mercado existen diversas tecnologías y herramientas que pueden acceder a la información y realizar su respectivo análisis.

La gestión de la información en cada empresa es diferente, al igual que en cada unidad

departamental de la misma. Cada gerente o jefe de área procesa la información de una forma, muchas veces personalizada. Y es por esta razón que, a veces, no logran llegar a un consenso en la toma de decisiones.

Con la velocidad cambiante de los mercados y de las tecnologías, es necesario industrializar la información para que se puedan tomar decisiones de una manera rápida y eficaz.

Para lograr el proceso de facilitar la información clave para la toma de las decisiones en una empresa se utiliza el término **BI**, por sus siglas en inglés Business Intelligence, que en español se traduce como Inteligencia de Negocio. Para efectos de este proyecto se utilizarán los tres términos. El Business Intelligence [3] tiene como objetivo poder ayudar a las empresas con su información y que logren una toma de decisiones más eficiente.

Si se analizan las diferentes metodologías y herramientas que existen en el mercado de Business Intelligence, dentro del testing, se tiene información suficiente para poder elegir la que mejor funcione, a modo de lograr un mayor control sobre los impactos de los cambios que se realicen en este entorno, reduciendo los tiempos de testing en los desarrollos de este tipo. Igualmente, utilizar la herramienta adecuada puede mejorar la calidad de los datos que se recopilan y su uso en futuros proyectos desarrollados en entornos **BI**.

El Business Intelligence permite observar cambios, tendencias y variabilidades en el tiempo si se hace un análisis continuo de la información. También se pueden utilizar estas herramientas para el análisis de información en un momento puntual. Cuando se aplica **BI** se obtiene una información estructurada, es decir, los datos son almacenados en tablas que se relacionan entre ellas. A su vez, estas tablas son almacenadas en bases de datos.

En todo proyecto de Business Intelligence se trabaja para conseguir un objetivo concreto ya definido. Por ejemplo, conocer el incremento de las ventas, la reducción de costos, el objetivo de ventas presupuestado, etc. Al obtener un resultado, este se comunica y se pueden realizar los cambios necesarios en la empresa para poder mejorar su competitividad. Para conseguirlo es necesario que los datos con los que se trabaja sean correctos y fiables. Por tanto, es necesario realizar una serie de comprobaciones que se realizan mediante el testing.

Al conjunto de datos con el que se trabaja se le deben realizar pruebas durante su implementación a lo largo de todo el ciclo de transformaciones, esto otorga confianza a los usuarios para la toma de decisiones. Mediante el testing se debe asegurar la integridad, la precisión, seguridad, consistencia y fiabilidad de los datos que se van a usar durante todas las etapas por las que pasan y no solo en los informes y dashboards que el usuario final va a utilizar para la toma de decisiones [4].

Se detecta la necesidad de emplear herramientas de testing **BI** que aceleren y den más confianza al usuario sobre los datos que se utilizan. Las herramientas de testing **BI** van a

aportar mayor fiabilidad a los datos y nos van a ayudar a reducir los tiempos empleados en el testing reportando mayor beneficio a las compañías.

El testing debería ser fundamental a la hora de desarrollar un proyecto BI y se deben realizar pruebas en todos los puntos de entrada de información a lo largo del ciclo de desarrollo [5].

Este trabajo se centrará en reforzar esta idea y comprobar de forma tangible el beneficio que puede reportar el uso de las herramientas testing durante el desarrollo de un proyecto BI.

1.1 Objetivos

Este proyecto se realiza con el objetivo de dar solución a los problemas detectados a la hora de realizar testing sobre las soluciones BI generadas para una determinada empresa, apoyándose sobre todo en el conocimiento de las tecnologías de la información y las herramientas existentes actualmente en el mercado.

1.1.1 Objetivo general

Consiste en un estudio de las herramientas, metodologías y estado del arte de la aplicación de testing en sistemas de Inteligencia Empresarial (BI), sus usos en un entorno real y determinar cuál es la mejora que puede generar el uso de estas.

Además, se realizarán pruebas con un modelo piloto formado por datos reales, donde se pueda observar, de forma tangible, la mejora que supone el uso de estas herramientas sobre un caso de uso en un entorno real.

1.1.2 Objetivos detallados

Para poder obtener el objetivo anteriormente planteado, primero se necesita abordar los siguientes puntos que nos ayudarán a alcanzarlo:

- Obtener mayor control sobre los impactos de los cambios que se realicen en los proyectos BI.
- Reducir los tiempos de testing en los desarrollos de proyectos de BI.
- Mejorar la calidad de los datos usados en BI.
- Mejorar la calidad de los desarrollos de proyectos BI.

1.2 Organización de la memoria

La memoria está formada por un total de ocho capítulos. A continuación, se describirá brevemente lo que se trata en cada uno de ellos:

- **Capítulo 1. Introducción:** En este capítulo se muestra una visión general del proyecto e indicando los objetivos que se quieren obtener.
- **Capítulo 2. Metodología:** En este capítulo se muestra la metodología utilizada durante el desarrollo del proyecto. Se realiza una descripción de la metodología y como se ha adaptado al proyecto. También se muestra la planificación realizada del proyecto y la estimación de costes.
- **Capítulo 3. Inteligencia de Negocio:** En este capítulo se describe qué es la Inteligencia de Negocio y algunas de sus posibles aplicaciones. Además, se muestran las características más comunes de estos sistemas y los componentes que los forman.
- **Capítulo 4. BI en las empresas:** En este capítulo se expone que se necesita a la hora de implementar un sistema Business Intelligence en las organizaciones. También se explica qué es la minería de datos y para qué se utiliza, así como los procesos que se realizan durante su aplicación.
- **Capítulo 5. Testing en BI:** En este capítulo se explica más en detalle en qué consiste el testing y que utilidades tiene su aplicación a mayores de los beneficios que aporta su uso. También se realiza una pequeña introducción de algunas de las herramientas de testing disponibles en el mercado actualmente.
- **Capítulo 6. Herramientas de Testing en BI:** En este capítulo se realiza una presentación más en detalle de las herramientas anteriormente mostradas en el capítulo 5. Se explica su uso, su aplicación y los beneficios que aporta su utilización en cada una de ellas.
- **Capítulo 7. Caso de estudio: NBI:** En este capítulo se realizan las pruebas con la herramienta NBI, que es la seleccionada en este proyecto, para comprobar su funcionalidad. Se realizan test sobre un conjunto de datos de un entorno real para medir su utilidad y documentarla.
- **Capítulo 8. Conclusiones y trabajos futuros:** En este capítulo se realizan las observaciones finales del proyecto y se describen las resoluciones alcanzadas. También se describe el grado de cumplimiento de los objetivos propuestos. Se incluye una pequeña sección en donde se indica el trabajo a posterior tras el desarrollo del proyecto.

- Anexo **A. Guía de instalación**: En este anexo se ha redactado una guía de instalación de la aplicación NBI.
- Anexo **B. Test de prueba básicos con NBI**: En este anexo se han incluido algunos de los test creados más básicos; utilizados para entender el funcionamiento de la aplicación NBI.

Metodología

En este capítulo se describe la metodología utilizada para la creación del proyecto, así como, las adaptaciones realizadas para este desarrollo. En la sección 2.1 se explican las características principales de la metodología escogida. Posteriormente, en la sección 2.2, se muestra como ha sido la adaptación realizada al proyecto. Finalmente, en la sección 2.3 se describe la planificación y seguimiento del proyecto.

2.1 Metodología SCRUM

Para el desarrollo de este trabajo se ha optado por el uso de la metodología Scrum [6] [7]. Es una metodología ágil en la que las personas, equipos y organizaciones generan valor a través de soluciones adaptativas para problemas complejos. Esta metodología se basa en entregas parciales y regulares del producto final que se quiere conseguir priorizando el beneficio que aportan al receptor del proyecto. Scrum emplea un enfoque iterativo e incremental para optimizar la previsibilidad y controlar el riesgo.

En los siguientes apartados se explicarán los roles que puede desarrollar cada participante del proyecto, los documentos necesarios y el proceso a seguir.

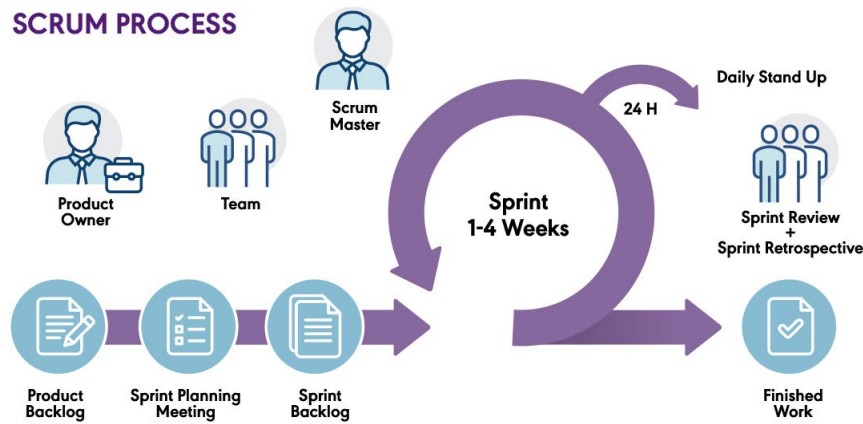


Figura 2.1: Metodología SCRUM

2.1.1 Roles

En la metodología Scrum se consideran tres roles:

- **Equipo de desarrollo:** Está formado por el conjunto de profesionales más técnicos que desarrollan el proyecto. Los integrantes del equipo de desarrollo pueden desarrollar múltiples funciones y se autoorganizan el trabajo ellos mismos, y no tienen jerarquías dentro del equipo.
- **Product Owner:** Sería representado por el cliente y las necesidades que presenta. Es el responsable de que exista una priorización de los objetivos a cumplir y maximizar el valor del producto y el trabajo del equipo. Es la persona responsable de gestionar el *Product Backlog*.
- **Scrum Master:** Su objetivo es conseguir que se esté trabajando de la forma más efectiva posible y se asegura de que el equipo aplica los principios de la metodología de forma correcta. También se ocupa de planificar y dirigir las reuniones y de solucionar los obstáculos que puedan presentarse.

Aunque oficialmente Scrum solo considera tres roles en su metodología, podría añadirse un cuarto rol que sería *Stakeholders*, el cual englobaría a todas las demás personas implicadas en el proyecto pero que no forman parte del equipo de Scrum. Son personas que tienen el conocimiento necesario a la hora de realizar el proyecto (*usuarios finales, equipo IT de cliente, gerentes...*) y son representados por el *Product Owner* dentro del equipo Scrum.

2.1.2 Actividades o eventos

En Scrum existen actividades o eventos predefinidos y son los siguientes:

- **Sprint Planning:** Es la reunión en la que se planifican las tareas a realizar durante el próximo *Sprint*. El *Scrum Master* se asegura de que se realice y que los asistentes entiendan la finalidad.
- **Sprint:** Es el evento principal. Se trata de un período de tiempo fijo (*normalmente entre una o dos semanas y un máximo de cuatro*), en el que el equipo de desarrollo trabaja en las tareas definidas para ese *sprint*. Es importante que durante la ejecución de los *Sprints* no se realicen modificaciones en los objetivos del mismo. Cada *Sprint* nuevo comienza inmediatamente después de la finalización del anterior.
- **Scrum Daily Meeting:** Es una reunión diaria que se establece con un máximo de quince minutos para que el equipo de desarrollo se ponga al día de qué tareas está haciendo cada uno, la previsión de tiempo que les va a llevar terminarlas y se resuelven dudas compartiendo conocimientos e información útil para el proyecto. Esta reunión la convoca el *Scrum Master* pero es el equipo de desarrollo el encargado de gestionarla durante su ejecución.
- **Sprint Review:** Se realiza una vez finalizado el *Sprint*. Se trata de una reunión informal y se realiza una presentación de las tareas realizadas durante el *Sprint* que tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración entre el equipo Scrum.
- **Sprint Retrospective:** Es una reunión que se lleva a cabo al finalizar la revisión del *Sprint* y antes de la planificación del siguiente. El *Scrum Master* debe motivar al equipo para que mejore, revisando cual fue el resultado del último *Sprint* e identificar y ordenar las tareas más importantes que finalizaron correctamente y las posibles mejoras a aplicar en los próximos *Sprints*.

El *Product Owner* es el encargado de indicar qué elementos del *Product Backlog* han sido completados y cuáles no.

2.1.3 Herramientas o artefactos

Las herramientas o artefactos de Scrum son la información que utiliza el equipo para desarrollar un proyecto, las actuaciones que deben llevar a cabo para realizarlo y todas las tareas que se deben realizar durante el proyecto para poder completarlo satisfactoriamente. Más en detalle son:

- **Product Backlog:** Es una lista dinámica priorizada de los requisitos u objetivos expresados por el cliente respecto a las entregas del proyecto. El *Product Owner* es el responsable de la lista, incluyendo la gestión de su contenido, disponibilidad y ordenación. En la lista se pueden marcar fechas de entrega esperadas por el cliente.
- **Sprint Backlog:** Es el conjunto de elementos del *Product Backlog* seleccionados para realizar en el *Sprint*. Esta lista es una *foto* en tiempo real del trabajo que el equipo de desarrollo pretende ejecutar durante el *Sprint*.
- **Burndown charts:** Es un gráfico que nos muestra a la velocidad a la que se están logrando los objetivos. Sirve para realizar simulaciones de tiempo al aplazar fechas de entrega y también se usa para estimar si el proyecto finalizará en el tiempo proyectado.
- **Definition of Done:** Busca que al finalizar el *Sprint* tengamos un proyecto *potencialmente entregable y usable* y ayuda a establecer un criterio de calidad mínimo que se debe cumplir en cada iteración para lograrlo.

2.2 Adaptación de la metodología Scrum al proyecto

Al tratarse de un *Trabajo de Fin de Grado*, se ha tenido que adaptar la metodología Scrum a las características del mismo. Dentro de este proyecto, funcionaría de la siguiente manera tras la adaptación:

La definición de los roles es distinta a la marcada por la metodología y, se organiza de la siguiente forma:

- **Product Owner:** El rol será desempeñado por los dos directores del trabajo y por el tutor de la empresa colaboradora.
- **Equipo de desarrollo:** El rol será desempeñado la autora del TFG.
- **Scrum Master:** El rol será desempeñado la autora del TFG.

Para este proyecto, la actividad de *Scrum Daily Meeting* del equipo no aplica al tratarse de un equipo formado por una única persona.

2.3 Gestión del proyecto

En esta sección, se exponen los recursos necesarios para realizar este proyecto, así como las estimaciones del tiempo y costes realizadas al inicio.

2.3.1 Planificación y estimación del proyecto

Dado que el proyecto consta de una parte teórica y una parte práctica, la forma en la que se ha organizado la planificación y realización del trabajo ha sido la siguiente:

1. **Planificación de las tareas:** Se ha creado un listado con todas las tareas a abordar para la correcta realización del proyecto y se han repartido en *Sprints* para abordarlas correctamente en tiempo y forma según prioridad.
 - **Búsqueda de documentación y bibliografía.**
 - **Formación sobre las herramientas y tecnologías que se van a emplear en el proyecto.**
 - **Selección de la herramienta que se va a utilizar para el caso práctico:** Una vez seleccionada se definen los casos de uso y el o los test a realizar, se analizan los resultados y se comparan los resultados.
2. **Revisión de las conclusiones obtenidas.**
3. **Redacción de la memoria y la documentación asociada al proyecto.**
4. **Revisión y correcciones.**

2.3.2 Estimación

Para este trabajo se ha proyectado una duración de once semanas, con un trabajo total máximo por semana de doce horas repartidas de lunes a sábado. El proyecto se ha dividido en nueve *Sprints* con una duración acotada de una semana cada uno, excepto para los *Sprints* siete y ocho, que tienen una duración excepcional de dos semanas para poder cubrir el tiempo necesario para desarrollar las tareas.

Para poder visualizar de forma rápida y sencilla el trabajo y cómo se ha distribuido en el tiempo, se elabora un diagrama de Gantt. El diagrama de Gantt [8] es una herramienta utilizada en la planificación y gestión de proyectos que sirve para ver las tareas y su programación en el tiempo de forma rápida mediante un gráfico de barras.

En la figura 2.2 se muestra el diagrama de Gantt que representa el trabajo realizado en este proyecto.

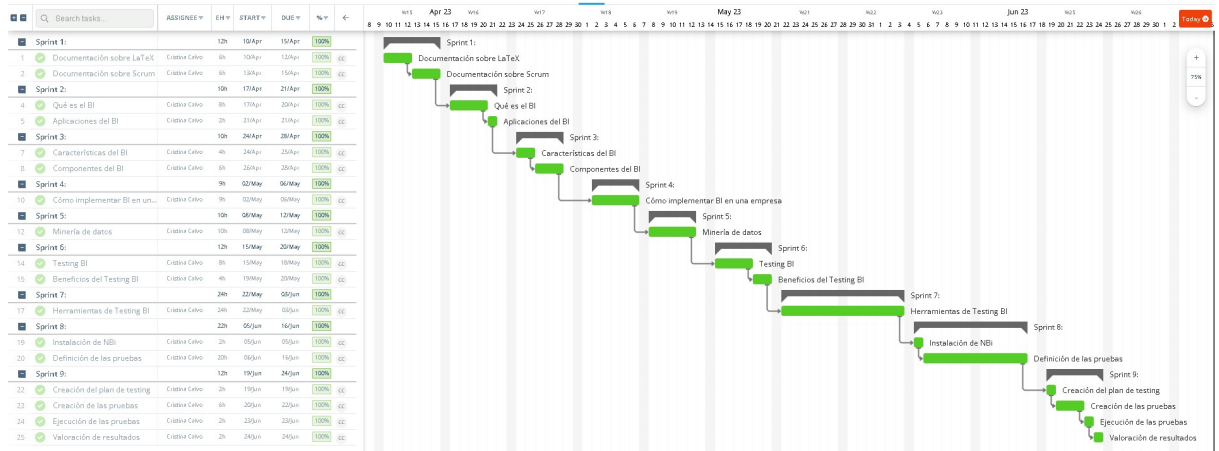


Figura 2.2: Diagrama de Gantt. Planificación del proyecto

A continuación, se explica el trabajo planificado para cada uno de los *Sprints* que se puede ver en la figura 2.2:

- *Sprint 1*: Para este *Sprint*, se planifican las tareas de estudio de la documentación de la metodología Scrum y la documentación de LaTeX para su posterior aplicación en este trabajo.
- *Sprint 2 y 3*: Estos *Sprints* se van a dedicar a las tareas de búsqueda de conocimiento e información sobre qué es la Inteligencia de Negocio, qué aplicaciones puede tener dentro de las empresas, así como sus principales características y componentes. Se han creado cuatro tareas repartidas entre ambos *Sprints* para poder abarcarlas.
- *Sprint 4 y 5*: Para estos dos *Sprints* se han definido las tareas de implementación de la Inteligencia de Negocio dentro de una compañía, cómo se debe implementar, qué estrategias y modelos se pueden seguir, así como los niveles de implementación de BI que existen dentro de una compañía. También se planifican las tareas de recopilación de información sobre qué es la minería de datos y que procesos se realizan durante su uso.
- *Sprint 6*: Este *Sprint* se dedica a las tareas de búsqueda de información sobre el Testing y los beneficios que puede aportar su uso.
- *Sprint 7*: Este *Sprint* tiene una duración excepcional de dos semanas para poder realizar todas las tareas planificadas. En él, se va a realizar un estudio sobre las herramientas más relevantes para el Testing BI. Una vez identificadas las herramientas, se procede al estudio pormenorizado de cada una de ellas para valorar su funcionamiento, uso, coste y beneficios que puede aportar cada una de ellas al caso de estudio.

- *Sprint 8*: Este *Sprint* también tiene una duración excepcional de dos semanas. Se plantean las tareas de instalación de la herramienta seleccionada y la realización de varias pruebas con una *BB.DD.* libre para la primera definición de las pruebas a implementar para el caso de estudio.
- *Sprint 9*: Las tareas definidas para este último *Sprint* son la creación de la planificación de las pruebas, la creación de pruebas específicas para el caso de estudio, tanto con la herramienta como para la validación manual; su ejecución y posterior análisis de los datos obtenidos tras su realización.

2.3.3 Costes

Para estimar el coste del proyecto, es necesario plantear el coste de los recursos humanos (en este proyecto, el desarrollador) que realizan el proyecto y el coste material de las herramientas necesarias para su desarrollo.

En el caso de los recursos materiales, la herramienta seleccionada es una herramienta *OpenSource*, por lo que su coste para el proyecto será de 0€. Además, necesitaremos el software *Visual Studio*, pero podremos realizar el proyecto con la licencia gratuita, por lo que su coste también será de 0€ para el proyecto. Debemos también tener en cuenta el coste del equipo hardware que se necesita para su realización, en este caso solo necesitamos el uso de un ordenador personal. El coste de los recursos materiales se detalla en la tabla 2.1.

Recurso	Unidades	Coste (€/ud)	Vida (meses)	Uso (meses)	Total (€)
<i>Ordenador personal</i>	1	900	48	6	112,50
<i>Herramienta NBi</i>	1	0	Ilimitado	6	0
<i>Licencia Visual Studio</i>	1	0	Ilimitado	6	0
Total					112,50

Tabla 2.1: Coste materiales

Mientras que el coste del equipo desarrollador resultará ser las horas empleadas en el proyecto. Para especificar los costes relativos a los recursos humanos se hace uso del *XX Convenio colectivo nacional de empresas de ingeniería; oficinas de estudios técnicos; inspección, supervisión*

y control técnico y de calidad [9]. En una estimación donde el desarrollador es un consultor BI, teniendo en cuenta el convenio y el precio actual de mercado, su precio por hora se estimará en 55€ para el cliente. Por lo que el coste del consultor será $121h \times 55€/hora = 6.655€$. El coste de los recursos humanos se muestra en la tabla 2.2.

Rol	Tiempo (h)	Coste (€/h)	Total (€)
<i>Desarrollador</i>	121	55	6655,00
Total			6.655,00

Tabla 2.2: Coste recursos humanos

Por tanto, el coste total del proyecto se refleja en la tabla 2.3. Sumando todos los recursos empleados en él, asciende a **6.767,50€**.

Recursos	Coste (€)
<i>Humanos</i>	6655,00
<i>Materiales</i>	112,50
Total	6.767,50

Tabla 2.3: Coste total

2.3.4 Gestión de riesgos

En un proyecto es importante identificar los posibles riesgos a los que puede estar expuesto durante su elaboración.

En este proyecto, uno de los principales riesgos que se detecta es que pueda existir una mala planificación del tiempo y una estimación errónea de las tareas, lo que provoque un retraso en las entregas acordadas. Para evitarlo, se han estimado las tareas con un pequeño margen y se ha intentado ajustar lo máximo posible el trabajo realizado al planificado.

Otro riesgo importante que se detecta es una divulgación o acceso no autorizado a la información del cliente. Para evitarlo, se han enmascarado los datos durante la redacción de la memoria y los ejemplos usados se han realizado con una BB.DD. gratuita de *Microsoft*.

Por último, otro gran riesgo a tener en cuenta es una mala ejecución de las pruebas y, por tanto, una mala calidad de los datos e información. Para evitarlo, se diseña un plan de acción en donde se definen los objetivos que se quieren alcanzar, así como los procedimientos y pruebas que se deben realizar para obtenerlos.

Inteligencia de Negocio

En la actualidad el término Inteligencia de Negocio o Business Intelligence se ha hecho muy popular, sin embargo, sus orígenes son de hace bastante tiempo. En este capítulo veremos más en detalle qué es el **Business Intelligence (BI)**, qué componentes los forman, así como qué beneficios aporta su uso dentro de las empresas.

3.1 Definición de Inteligencia de Negocio o Business Intelligence

En el año 1958 fue Hans Peter Luhn, investigador de IBM, quien presenta el término Business Intelligence por primera vez. Conceptuándolo de la siguiente manera: *“es la habilidad de aprender las relaciones de hechos presentados de forma que guíen las acciones hacia una meta deseada”* [3].

Es un concepto bastante sencillo que abre las puertas a lo que es hoy la Inteligencia de Negocio o Business Intelligence. Pero fue en 1989 que el concepto es presentado formalmente por Howard Dresden, analista de la empresa **Gartner**, como: *“Conceptos y métodos para mejorar las decisiones de negocio mediante el uso de sistemas de soporte basados en hechos”* [3].

Sin embargo, en el glosario de términos de Gartner ¹ el término Business Intelligence se expresa de la siguiente manera: *“BI es un proceso interactivo para explorar y analizar información estructurada sobre un área (normalmente almacenada en un Datawarehouse), para descubrir tendencias o patrones, a partir de los cuales derivar ideas y extraer conclusiones. El proceso de Business Intelligence incluye la comunicación de los descubrimientos y efectuar los cambios. Las áreas incluyen clientes, proveedores, productos, servicios y competidores.”* [10].

En resumen, los proyectos de **Business Intelligence (BI)** utilizan la tecnología para analizar

¹ **Gartner**: Es una consultora internacional especializada en Tecnologías de Información y Comunicación, www.gartner.com, enero 2006

la información que posee la empresa basándose en un objetivo concreto. Luego, los resultados de ese análisis muestran lo que se ha descubierto durante el proceso, y sirve para que los directivos, o personas autorizadas en la toma de decisiones, puedan definir los cambios que deben realizarse en la empresa para lograr los objetivos propuestos.

Esto quiere decir que, por ejemplo, si una empresa necesita alcanzar una meta específica en ventas de un producto en sus tiendas, se recopila la información que se va a analizar. Esta información se encuentra en tablas que, a su vez, se encuentran en un repositorio como una base de datos. En BI se utilizan herramientas tecnológicas que facilitan este análisis y arrojan un resultado que corresponde con el objetivo planteado del aumento de las ventas de un producto específico.

En el resultado se muestran todas las variables encontradas, por ejemplo, que el producto se vende mejor en unas tiendas y en otras no, cuáles son las variables que han intervenido para que esto suceda: promoción del producto, ubicación, ofertas, coste, tipo de cliente, etc. y, las personas encargadas de tomar las decisiones determinan los próximos pasos a realizar en la empresa para lograr que el producto aumente sus ventas en las tiendas que tienen poca salida.

Aunque estas decisiones no siempre serán iguales, quizás en una tienda el producto tiene pocas ventas porque está ubicado en un lugar de poca afluencia, por lo que no lo pueden ver con facilidad. Mientras que en otra tienda el motivo será porque no le dan la suficiente promoción y los clientes no son conscientes de que allí se vende ese producto. Aunque en ambos casos el resultado es el desconocimiento sobre el mismo por parte de los clientes, las decisiones a tomar serán diferentes en cada caso.

Las respuestas encontradas tendrán un impacto importante en las ganancias y pérdidas de la empresa y en el desarrollo de nuevas estrategias. Sin embargo, es un análisis continuo después de haber implementado Business Intelligence, ya que no se va a implementar un proyecto de este tipo cada vez que se necesite analizar la información. Es por esta razón que la empresa debe contar con un sistema automatizado para esta función.

La implementación de un proyecto de BI no es una tarea inmediata, es un proceso complejo que requiere de un amplio conocimiento del entorno de la organización. Sin embargo, los resultados obtenidos otorgan grandes beneficios a la empresa.

3.2 Áreas de aplicación

Los proyectos de BI pueden ser aplicados en diferentes áreas de la empresa. Sus inicios fueron en ventas, pero ahora se pueden implementar en diversas áreas, y los responsables de

cada una tienen sus propios objetivos, a saber:

- **Compras:** se pueden analizar las tendencias de compras de los productos, cuáles son los que más se venden.
- **Ventas:** colocación de los productos, cuáles tienen una alta rotación y cuáles no, de esa manera se pueden ubicar en zonas preferenciales. Reponer productos con mayor o menor frecuencia.
- **Negociación:** los responsables de esta área pueden acordar tratos con las entidades financieras para conocer el flujo del dinero en tarjetas de crédito, débito o efectivo.
- **Marketing:** conocer la eficacia de las promociones para crear nuevas estrategias de marketing y atraer mayor público.
- **RR.HH.:** contratar mayor personal de acuerdo a la afluencia de los clientes en distintas épocas del año, asignar turnos.

Destacar que también es necesario un proyecto de BI en todas aquellas áreas que utilizan demasiado Excel para el análisis de la información, donde esta información no está centralizada y no se registra en el sistema de gestión de la empresa, lo cual es un problema. Esto es lo que se llama "Excel caos"².

De acuerdo con el objetivo que se tenga en la empresa, el cual puede ser de tipo operativo o estratégico, se establecerá el modelo de BI correspondiente y las personas responsables de tomar las decisiones, personas de negocio, serán a las que se les entregarán los resultados para su análisis y toma de decisiones. Por supuesto, toda esa información debe ser entregada por las personas que la manejan y debe estar actualizada.

Las personas deben tener clara la importancia del Business Intelligence. La gran mayoría de las empresas tienen personal preparando información por un período de tiempo muy extenso, mientras que al análisis solo le dedican un tiempo muy corto. La toma de decisiones es muy importante dentro de una empresa, por lo que se debe dedicar más tiempo en analizar la información. Muchas veces son grandes volúmenes de información para analizar, sin embargo, con BI se pueden manejar estos volúmenes de forma automatizada, mejorando los procesos y facilitando la toma de decisiones.

² Excel Caos: Problema resultante del uso excesivo de Excel como herramienta de análisis. Al tener cada usuario un fichero personalizado, existe un coste muy alto al querer sincronizar todos los ficheros para que cuadren y se puedan utilizar en la toma de decisiones.

3.3 Características

Al implementar un proyecto de BI se utilizan diversos software y herramientas que automatizan el análisis de la información. Mientras que, con los sistemas manuales, como hojas de Excel, se necesita demasiado tiempo para recopilar la información y luego procesarla, quedando muchas veces demasiada información pendiente de analizar, por lo que no se logra tomar las mejores decisiones.

Estos sistemas BI tienen características [11] que son comunes entre ellos, por ejemplo:

- Están orientados a un consumidor final, es decir, son software fáciles de utilizar por cualquier persona. El usuario no necesita contar con una especialización para poder manejarlos. Además, son amigables e intuitivos.
- Garantizan la accesibilidad de la información. Esta se encuentra disponible sin importar su procedencia o formato.
- Los informes generados permiten que las personas claves en la empresa puedan tomar decisiones importantes basadas en datos. De esta manera se puede mejorar el posicionamiento de la empresa. Las herramientas utilizadas permiten que se pueda seleccionar la información de acuerdo con la necesidad y el interés en ella.
- Existen versiones móviles de estas herramientas, lo que facilita la consulta de los datos, la creación del análisis de estos y el seguimiento del proyecto por parte de las autoridades de la empresa, entre otras características.

Otras características que identifican a los proyectos de BI son:

- **Arquitectura abierta:** existen soluciones OpenSource de los software y herramientas BI que permiten una mayor compatibilidad con los sistemas de las empresas y pueden integrarse fácilmente.
- **Mayor cantidad de tipos de bases de datos soportadas:** la mejor alternativa en BI es aquella que puede soportar una mayor amplitud en las bases de datos. Las bases de datos son cruciales en la implementación de un proyecto BI, debido a que son las que contienen gran cantidad de datos. Por esta razón, deben garantizar la integración con la base de datos de la empresa.
- **Datos en tiempo real:** muchas veces es necesaria la toma de una decisión con carácter de urgencia. Es por eso por lo que un sistema de BI idóneo es aquel que puede suministrar esos datos en tiempo real. Esto agiliza la toma de las decisiones.

- **Compatibilidad con Datawarehouse y Datamart:** es necesario que los datos a utilizar se encuentren en un mismo lugar y que cuenten con un mismo formato. Esta estructura se mantiene en un Datawarehouse o Datamart que es donde se almacenan los datos. De esta manera, el procesamiento y análisis de la información es mucho más ágil y eficaz.
- **Autoservicio:** esta capacidad de los sistemas de BI permite que los usuarios puedan crear sus propios informes, ejecutar consultas personalizadas y determinar los patrones a utilizar para el análisis de la información.
- **Versión móvil:** en la actualidad cualquier aplicación debe tener su versión móvil debido a la movilidad de las personas, el trabajo desde casa, el uso de las tabletas y los smartphones. Los sistemas de BI no escapan de esta realidad.
- **Dinamismo:** al obtener información de alto valor, las empresas pueden analizar su estado actual en cualquier momento.
- **Software intuitivo:** los software utilizados en un proyecto de BI son intuitivos y visuales, generan gráficos sencillos que pueden ser interpretados con gran facilidad por los usuarios finales.
- **Visualización de métricas:** las métricas se obtienen del análisis de una gran cantidad de información en un tiempo muy corto comparado con el procesamiento manual de datos. Los indicadores con los que se trabaja resumen toda la información de manera automatizada.
- **Acceso a la información constante:** a través de internet se puede tener acceso a la información de manera constante e inmediata. Esto permite que se puedan tomar decisiones críticas en cualquier momento y lugar.

3.4 Componentes

Para que el sistema de BI funcione se realizan diversos procesos, por ejemplo, se necesita ubicar los datos de diferentes fuentes, extraer los datos que se requieren, procesarlos y transformarlos a un mismo tipo, almacenarlos en un destino temporal donde se procesarán y luego pasar los resultados al destino final. Desde allí, pueden ser consultados desde cualquier parte con distintos dispositivos. A grandes rasgos, podría parecer un proceso sencillo, pero se necesita invertir tiempo y debe ser muy detallado.

En las siguientes secciones, se detallan los pasos que siguen los datos en un sistema BI [12]. Estos pasos y los componentes implicados en cada uno de ellos se ilustran en la figura 3.1.

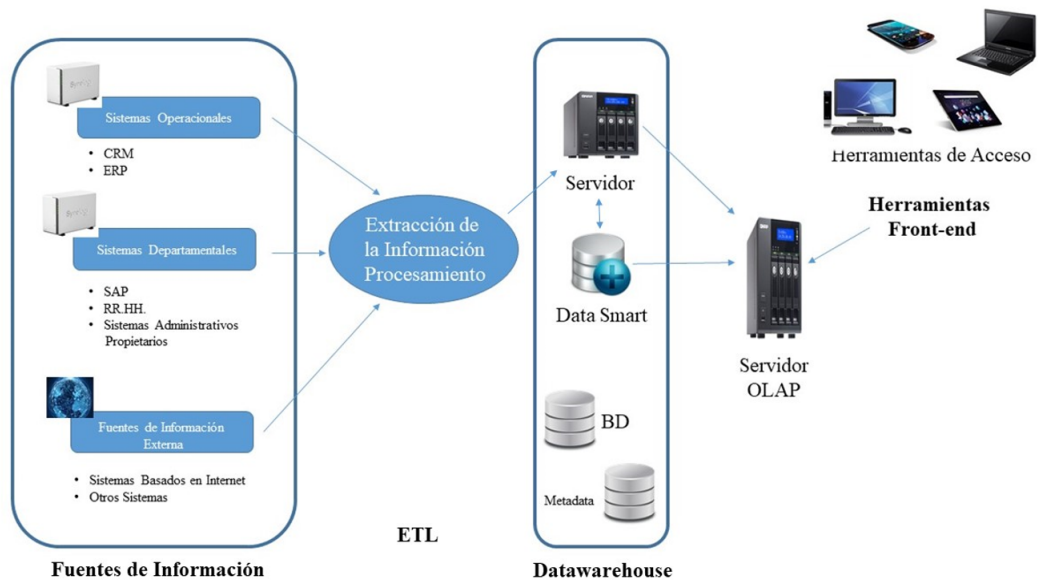


Figura 3.1: Componentes de Business Intelligence

3.4.1 Fuentes de información

Son todas las fuentes de donde se obtienen los datos que se necesitan para cumplir el objetivo propuesto y poder ayudar en la toma de las decisiones en algún campo específico.

Estas fuentes de datos pueden ser de origen interno o externo. Las de origen interno se refieren a aquellos datos que se encuentran en diferentes formatos dentro de la propia empresa, ya sean hojas de cálculo de Excel o bases de datos. Es decir, son datos que se extraen de sistemas que funcionan en la empresa, sean estos operacionales o departamentales, tales como: CRM, ERP, RR.HH., otros sistemas contables, etc. Mientras que las fuentes de datos externas se extraen de fuera de la empresa, tales como las redes sociales o fuentes de datos abiertas (*Opendata*).

3.4.2 ETL

Después de identificar todas las fuentes de donde se extraerán los datos necesarios, los datos pasan por el proceso de ETL (*Extract, Transform, Load* que se traducen como *Extracción, Transformación y Carga*).

Es decir, como primer paso se inicia la extracción de los datos desde las diferentes fuentes seleccionadas. Luego se transforman estos datos en un formato único, ya que los datos proceden de fuentes diferentes que usan formatos diferentes, de esta manera se pueden trabajar sin inconvenientes. También se realiza la limpieza de los datos, la integración de estos y la

actualización.

Por último, se cargan los datos en el Datamart o Datawarehouse donde posteriormente se realizará su análisis.

3.4.3 Datawarehouse y/o Datamart

Como su nombre en inglés lo indica, un Datawarehouse es un almacén de los datos de la empresa, en cambio, el Datamart se utiliza sobre una unidad o departamento específico de esta. El Datawarehouse se trata de una base de datos más compleja, ya que está diseñada para optimizar y organizar los datos resultantes de la ETL, para su posterior análisis.

El Datamart es una base de datos más pequeña, el cual se encuentra en un entorno dentro de un almacenamiento mucho mayor como lo es el Datawarehouse. Se centran en los datos departamentales (*Logística, RR.HH., Marketing, Contabilidad, etc.*), a los cuales se puede acceder para realizar informes o análisis específicos.

En los Datawarehouse y/o Datamarts se pueden ejecutar análisis complejos y con grandes volúmenes de datos provenientes de diferentes fuentes. Aquí es donde realmente se analiza la información, de acuerdo con el objetivo que se persigue, y con los resultados obtenidos es que se pueden tomar decisiones importantes.

Los Datawarehouse y/o Datamarts tienen gran capacidad de almacenamiento y procesan grandes volúmenes de información que provienen de diferentes departamentos de la empresa y/o fuentes externas.

3.4.4 Servidor OLAP

Las siglas OLAP provienen del inglés *Online Analytical Processing*, lo que significa Procesamiento Analítico Online. OLAP representaría el front-end del sistema de Business Intelligence. Después de que todos los datos han pasado por un proceso ETL y se almacenan en el Datawarehouse, estos están disponibles para los usuarios avanzados de BI.

Es una herramienta muy poderosa y práctica para los analistas, ya que se pueden elegir datos, agruparlos y ordenarlos de manera que puedan dar respuesta a preguntas específicas.

3.4.5 Herramientas de visualización

Estas herramientas son las que los usuarios finales utilizan para visualizar los resultados del análisis de los datos.

En los Dashboards o Cuadros de mando se suelen visualizar mapas interactivos, los famosos *Key Performance Indicator (KPI)*, que son los indicadores de rendimiento y gráficos de diferentes tipos.

Los Dashboards se deben diseñar con los requerimientos del usuario final y mostrar la información de manera que sea fácilmente entendible y fiable. Por lo general, se realizan como un resumen que contiene los indicadores más relevantes en la pantalla principal y subpestañas a las que se puede acceder a través de los indicadores principales para aumentar el nivel de detalle de la información. El usuario final será el encargado de navegar por el dashboard y analizar la información obtenida con el análisis del informe.

Si se tiene un **Cuadro de Mando Integral (CMI)** [13] o también conocido como **Balanced Scorecard (BSC)** sería de mayor ayuda, ya que este es más completo y reúne los indicadores más relevantes que están orientados a los objetivos que, a su vez, se encuentran enlazados entre ellos. De esta manera, los resultados se orientan a los planes de acción propuestos por las estrategias establecidas por la empresa. A estos cuadros de mando se puede acceder a través de internet desde cualquier parte, siempre y cuando sea un usuario autorizado para ello. Igualmente, puede utilizar cualquier tipo de dispositivo, sea este un ordenador, tableta o teléfono. Esto facilita la toma de decisiones en momentos críticos. Durante el diseño del sistema de Business Intelligence, la creación del Dashboard no debe llevar más del 20% del tiempo y dedicación total del proyecto.

3.5 Beneficios de la Inteligencia de Negocio

La Inteligencia de Negocio brinda a las empresas grandes ventajas competitivas, gracias a que con ella se pueden tomar decisiones objetivas y con una base sólida en un tiempo menor.

Las empresas que utilizan proyectos de BI obtienen mayores ventajas frente a las que no. De esta manera, pueden implementar mejor sus estrategias, alinear los objetivos de cada departamento con los de la empresa y actuar a tiempo en casos que pueden volverse críticos.

Entre los múltiples beneficios [14] que se logran con un proyecto de Business Intelligence se encuentran los siguientes:

- **Homogeneización de los datos:** muchas veces para la toma de decisiones que afectará a la empresa, es necesario recopilar los datos de varios departamentos como Marketing, Compras, Ventas, etc., que se pueden encontrar almacenados en diferentes formatos. Con un proyecto de BI se logra consolidar la información necesaria en un formato homogéneo que permite el análisis con mayor eficacia. Estos datos pueden provenir de fuentes tanto internas como externas y, sin embargo, se pueden analizar como si proviniesen de una única fuente.
- **Círculo virtuoso de la información:** los datos generan conocimiento al transformarse en información valiosa. Con esta información, se pueden tomar mejores decisiones y

alcanzar resultados más eficientes para la empresa. Al mismo tiempo, se pueden generar nuevos datos que ayudarán a encontrar nuevas perspectivas dentro de la empresa.

- **Visión única de la información:** se puede obtener una visión más conformada, de calidad, persistente e histórica de la información que se maneja en la empresa. Esto facilita la toma de decisiones dentro de la organización o dentro de un departamento específico.
- **Elaboración de informes:** se facilita la elaboración de informes más exhaustivos utilizando grandes volúmenes de datos. Los datos son accesibles de forma rápida e intuitiva, lo que facilita que se puedan crear informes más detallados y útiles.
- **Métricas:** se pueden crear y mantener métricas con indicadores claves de rendimiento y de metas, los cuales son fundamentales para toda empresa. Esto también mejora el rendimiento de los departamentos, al manejar las métricas que le indicarán cómo es el avance de este dentro de un tiempo establecido.
- **Información actualizada:** la información obtenida en los informes estará actualizada, sin demoras ni errores, ya sea a modo general o detallado.
- **Mejora de la competitividad:** la competitividad de la empresa es mejorada gracias a que, con un proyecto de BI, se puede hacer una diferenciación de lo que es superfluo de lo que sí es relevante, se accede a la información más rápido y con mayor precisión, y la directiva de la empresa podrá tomar decisiones con mayor rapidez y de forma segura.
- **Identificación de puntos débiles:** al poder realizar un análisis de la información con más detalle, donde convergen los datos de varios departamentos, se pueden identificar las debilidades de la empresa y de esa manera, se pueden tomar acciones para corregirlas, mejorando la eficiencia operativa, reduciendo los costos, optimizando los procesos y, en consecuencia, aumentando los ingresos.
- **Control de la empresa:** al obtener una perspectiva más amplia de los procesos, procedimientos y la empresa en general, se puede ejercer un mayor control sobre la misma al tomar decisiones más fundamentadas.
- **Conocimiento de los consumidores y mercados:** se pueden conocer las tendencias de los mercados y las necesidades de los consumidores. Al tener este conocimiento, la empresa puede crear nuevas estrategias que se adapten a las tendencias y cumplan las expectativas de los clientes, aportándoles una mayor satisfacción y mayor valor a la empresa. Los informes generados darán un conocimiento más detallado y preciso del comportamiento de los clientes al tener la ventaja de poder manejar una gran cantidad de datos.

- **Detección de deficiencias:** las situaciones laborales de los empleados también se pueden analizar con un proyecto de BI, y pueden detectar a tiempo las deficiencias que existan. De esta manera, la empresa tiene la oportunidad de aplicar estrategias que busquen mejorar esas deficiencias y aumenten la motivación y el nivel de satisfacción del empleado, lo que conlleva un mayor compromiso de este con la empresa.
- **Capacidad de planificación:** otro de los beneficios que se obtienen con la Inteligencia de Negocio es mejorar la capacidad de planificación de los distintos departamentos de la empresa, gracias a la información obtenida. Al poder realizar pronósticos con una mayor precisión se logra una mejor organización del trabajo, de las estrategias y de los procesos.
- **Adaptable a cualquier negocio:** el BI se puede adaptar a cualquier tipo de negocio. Lo importante es saber interpretar los datos y utilizar la información resultante del análisis de estos de la mejor manera posible, facilitando una toma de decisiones más eficaz.
- **Orientada al usuario final:** es una herramienta intuitiva y fácil de utilizar, no se necesitan conocimientos profundos de análisis de datos. El usuario podrá crear informes que le ayuden a tomar decisiones rápidamente.

BI en las empresas

En este capítulo, nos centramos en las necesidades y requisitos que se presentan al implementar un sistema BI en una organización, qué es la minería de datos y qué se espera conseguir mediante su uso.

4.1 Implementación de BI en las empresas

Para implementar un proyecto de Business Intelligence en una empresa es necesaria una buena estrategia. Se deben combinar muy bien el uso, los procesos de madurez y las tecnologías, de manera que se coordinen de forma efectiva.

Para que funcione correctamente, se necesita una inversión de tiempo a largo plazo. La Inteligencia de Negocio va a ayudar a conocer el negocio y su funcionamiento, apoyando oportunamente en la toma de decisiones. En algunos casos se utilizará para solucionar problemas, en otros optimizará procesos; y en muchos otros se podrán generar estrategias inmejorables mediante el correcto uso de la Inteligencia de Negocio.

Cuando en una empresa no existe una estrategia definida de BI, se pueden apreciar las siguientes situaciones o percepciones:

- El único culpable de todos los problemas de BI es el departamento de IT de la empresa.
- Para la directiva, un proyecto de BI es un gasto adicional sin fundamento.
- Los usuarios no aclaran las necesidades de los informes que se deben generar, por lo que IT debe preguntar a los usuarios cada vez que se necesite un informe.
- El helpdesk de la empresa es quien da soporte al sistema de BI.
- La gestión del rendimiento no se diferencia de la Inteligencia de Negocio.

- Es imposible medir el retorno de la inversión (ROI) como el uso del sistema de BI.
- Debido a que no existe un plan definido, no se puede contratar, retener o aumentar el equipo de Inteligencia de Negocio.
- Si el proyecto de BI tiene asignado un responsable funcional, este no es el idóneo. En muchos casos ni siquiera existe este rol.
- Una solución debe cubrir a múltiples departamentos, y sin la debida estrategia, esto no se cumple y existen las soluciones con repetición de funcionalidades.
- No existe un plan de formación en las herramientas que se van a utilizar.
- Los usuarios finales tienen dudas con los datos almacenados en el Datawarehouse.

Como se ha mencionado anteriormente, un proyecto de BI conlleva el desarrollo de una estrategia, la cual necesita se lleven a cabo múltiples actividades, entre las que se pueden mencionar:

- Con la presencia de un ejecutivo y analistas de negocio, se debe crear un centro de competencia de BI, en el cual se combinarán los conocimientos en estrategias, tecnologías, metodologías, etc. para compartir tanto los éxitos como los fracasos y poder aprender de ellos.
- Racionalizar las tecnologías, tanto las existentes como las futuras adquisiciones, estableciendo los estándares de BI dentro de la empresa.
- En los casos en que no es posible crear el centro de competencia de BI, se debe alinear a IT con el negocio, de manera que entienda las necesidades del mismo, y pueda aportar las soluciones más idóneas a necesidades en particular, así como a aquellas a futuro.
- Se deben identificar los procesos de negocio que necesitan trabajar continuamente con diferentes aplicaciones, de manera que no se generen silos de funcionalidad.
- Con el desarrollo de un framework de métricas se puede gestionar el rendimiento a nivel empresarial.
- Para añadir valor a las decisiones que se tomen es necesario incluir, en los informes, los resultados que arrojan las diferentes aplicaciones analíticas en los procesos del negocio.
- El portafolio de las soluciones actuales se debe revisar y evaluar en cuanto a las ventajas y los riesgos.

- Se debe considerar el retorno de inversión (ROI), dentro del período de un año, de aquellas inversiones tácticas. Igualmente, analizar los mercados, las soluciones y, si es posible, el hyper cycle ¹ de Gartner.
- Es importante revisar los casos de éxito de otras empresas para así tener conocimiento de lo que ha funcionado como lo que no y poder aprender de esas experiencias.

4.2 Modelo de Madurez de Business Intelligence

El *Business Intelligence Maturity Model (BIMM)* o Modelo de Madurez de BI "es un modelo de madurez que permite clasificar nuestra organización desde el punto de vista del grado de madurez de implantación de sistemas Business Intelligence en la misma (en relación directa con frameworks como COBIT²)" [3].

En este modelo se distinguen siete fases, a saber:

1. **No existe BI:** Los datos se encuentran en los diferentes sistemas operacionales de cada departamento o en otros soportes dispersos por toda la empresa. Las decisiones son tomadas basadas en la experiencia o en la intuición, pero no en datos concretos. No se ha detectado la necesidad del uso de una herramienta analítica ni del uso de los datos para tomar decisiones importantes.
2. **No existe BI, pero los datos son accesibles:** la toma de decisiones no se hace a través del procesamiento formal de los datos. No obstante, algunos departamentos o usuarios justifican sus acciones basándose en datos de calidad que se encuentran en formatos sencillos como hojas Excel u otro tipo de informes. Hasta este momento, se desconoce la existencia del BI como una manera de mejorar los procesos dentro de la empresa, aunque se empieza a intuir que pueda existir.
3. **Aparición de procesos formales de toma de decisiones basadas en datos:** el control de los datos lo lleva un equipo, el cual genera los informes basados en los datos que toman de los sistemas transaccionales. Estos datos no se encuentran almacenados en un Datawarehouse, no son modelados y tampoco han pasado el proceso de limpieza y clasificación antes del proceso.

¹ *Hyper Cycle*: Representación gráfica de la madurez, adopción y aplicación de negocio de una o varias tecnologías específicas.

² *COBIT: Control Objectives for Information and related Technology*: Es un conjunto de mejores prácticas para el manejo de la información que fue creado por Information Systems Audit and Control Association (ISACA) e IT Governance Institute (ITGI) en el año 1992.

4. **Datawarehouse:** se percibe el Datawarehouse como un repositorio de datos que es muy necesario debido al impacto negativo que han experimentado anteriormente. Sin embargo, los informes siguen siendo personales.
5. **Datawarehouse crece y el informe se formaliza:** al ver que el repositorio funciona, aparece la necesidad de que beneficie a todos en la empresa y se formalizan los informes corporativos. Se comienza a escuchar sobre **On-line Analytical Processing (OLAP)**, aunque solo algunos usuarios entienden todos sus beneficios.
6. **Despliegue de OLAP:** después de usar los datos en el Datawarehouse por un tiempo, comienzan a aparecer preguntas más sofisticadas, haciendo que los informes no sean suficientes para responder las mismas ni se puedan procesar grandes volúmenes de datos y crear informes más elaborados que puedan responder a las necesidades de la organización. Es cuando se despliega **OLAP**, el cual puede cubrir dichos perfiles.
7. **Formalización de Business Intelligence:** comienzan a aparecer nuevas necesidades de implementar otros procesos de Inteligencia de Negocio, tales como *Balanced Score-Card* o *Data Mining*, por mencionar algunos, y el procesamiento de los datos con calidad comienza a impactar en otros sistemas como **Customer Relationship Management (CRM)** o **Supply Chain Management (SCM)**. Se ha implementado una nueva cultura empresarial en donde los usuarios entienden la diferencia entre los sistemas antiguos y modernos, dando paso a la formalización del **BI**.

4.3 Minería de datos (Data Mining)

La minería de datos [15] es una herramienta utilizada por la analítica de negocio que se ha expandido a muchos ámbitos de la industria, de los negocios, etc. En Inteligencia de Negocio es muy utilizado este término para desarrollar los proyectos de **BI** en la empresa, partiendo de los datos que se han recogido, organizado y almacenado con anterioridad.

La minería de datos se puede definir como “*el proceso de extraer conocimiento útil comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos*” [16].

La principal función de la minería de datos es crear modelos con estos datos y de esta manera beneficiar a la empresa, ya que la toma de decisiones es mucho más sencilla con el uso de patrones. Es un proceso semiautomático o totalmente automatizado.

Todo el proceso utiliza técnicas estadísticas, matemáticas y de inteligencia artificial, entre otras. Consiste en extraer los datos, identificar cuáles son de utilidad y luego, transformarlos en conocimiento en forma de patrones. Por ejemplo, en ventas, se podrían extraer los datos

de un cliente específico de manera que se puedan analizar los hábitos de compra, reconocer los nuevos productos que posee, sus características y finalmente encontrar las mejoras que se pueden lograr.

En resumen, la minería de datos encuentra la información predictiva dentro de enormes bases de datos para contestar aquellas preguntas complicadas que antes debían ser respondidas de manera manual, tomando una gran cantidad de tiempo y esfuerzo y, en muchos casos, no logrando una respuesta adecuada para la toma de decisiones.

4.4 Procesos de minería de datos

Los procesos de minería de datos son muy variados y cada empresa puede elegir el que mejor le funcione. Entre ellos se pueden mencionar: *Catalyst (P3TQ)*, *Semma*, *KDD Process*, entre otros. No obstante, uno que es muy popular entre los desarrolladores, investigadores y usuarios es el llamado **CRISP-DM**³, Proceso Estándar de la Industria para la minería de datos.

El ciclo de vida de un proyecto de minería de datos es planteado por esta metodología a través de seis etapas interrelacionadas (ver figura 4.1) [17]. Estas etapas son las siguientes:

1. **Compresión del negocio:** Es la primera fase y se centra en comprender los objetivos que se busca alcanzar, así como las exigencias que va a presentar el proyecto desde el punto de vista de negocio.
2. **Compresión de los datos:** En esta fase se busca identificar la calidad de datos con los que se va a trabajar, se debe comprobar su exactitud, totalidad, oportunidad, relevancia, nivel de detalle y consistencia. Una vez comprobada la calidad, podemos empezar a obtener conocimiento y descubrir patrones e información oculta mediante los conjuntos y subconjuntos de los datos.
3. **Preparación de los datos:** En esta fase se van a realizar los trabajos de limpieza y preparación de datos, los cuales pueden ser realizados varias veces para garantizar que los datos sean correctos.
4. **Modelado de los datos:** Una vez realizada la fase de preparación, se puede empezar a modelar los datos. Existen varias técnicas de modelado de datos, por lo que en esta fase debemos seleccionar la más adecuada de acuerdo con los objetivos que queramos obtener. Algunas de las técnicas necesitan aplicar unos requisitos específicos, por lo que puede ser necesario tener que volver a aplicar la fase de preparación para adaptar los

³ CRISP-DM: Cross-Industry Standard Process for Data Mining: Es una metodología estándar no propietaria que fue propuesta por un consorcio europeo a finales de los 90 para la implementación de proyectos de minería de datos.

datos. Algunas técnicas de modelado son: árbol de decisión, regresión lineal, clústering, reglas de asociación, modelado estadístico...

5. **Evaluación:** Una vez se ha realizado el modelado, debemos asegurarnos de que el modelo cumple con los estándares de calidad del proyecto. Es necesario realizar una serie de pruebas que nos ayuden a compararlo con los objetivos fijados en la primera fase y comprobar que cubre todas las expectativas de negocio.
6. **Despliegue:** Con el modelo ya validado se procede a la explotación y al uso de los datos obtenidos de él. Se pueden realizar desde informes hasta crear procesos cruzados de minería de datos entre áreas de la empresa. Normalmente esta fase la realiza una persona de negocio, la cual tiene el conocimiento suficiente para realizar las preguntas adecuadas al modelo.

La minería de datos es un proceso iterativo. El conocimiento obtenido del proceso de creación del modelo se puede utilizar como información de entrada y crear una nueva iteración que sirva para mejorarlo en un nuevo ciclo. Se pueden añadir nuevas fuentes de datos que aporten valor a la hora de obtener los objetivos, volver a transformar los datos, refinar el modelo, etc. En definitiva, tener creado un modelo no implica que se haya terminado el proyecto, solo que se ha terminado ese ciclo.

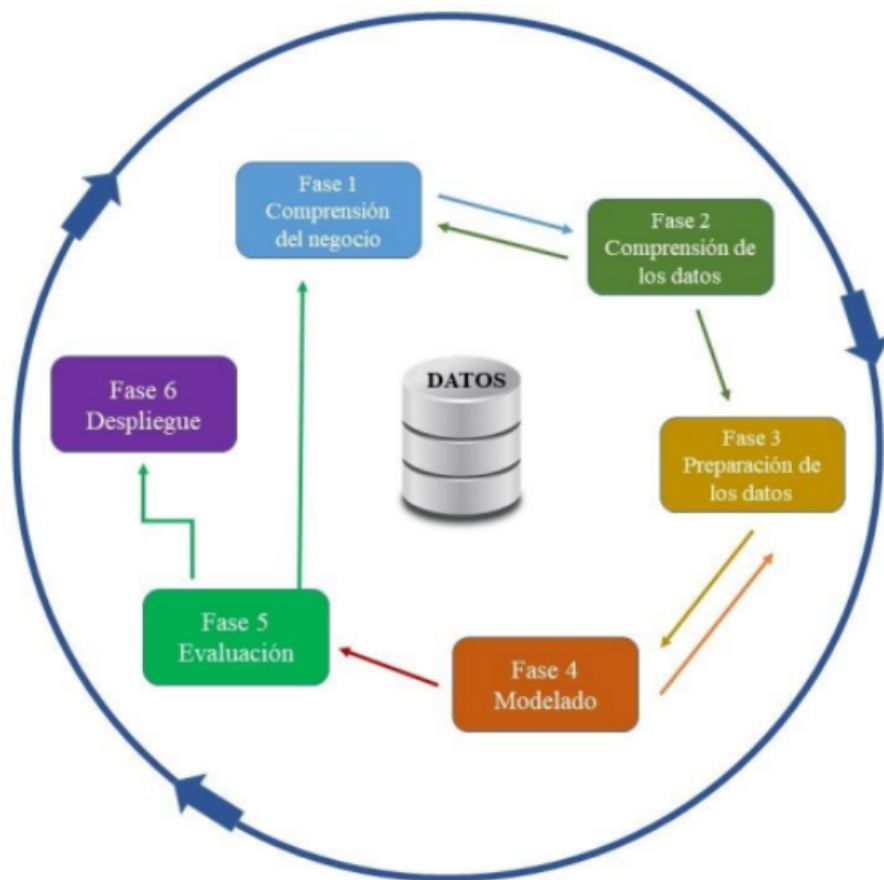


Figura 4.1: Metodología CRISP-DM.

Testing en BI

En este capítulo veremos más en detalle qué es el testing, para qué sirve y qué beneficios aporta su uso dentro de un proyecto Business Intelligence. Además, veremos un adelanto de las herramientas de testing que se analizarán más adelante en este proyecto.

5.1 Definición de Testing

Testing [18] es una palabra del inglés cuyo significado es pruebas. Y como su nombre indica, son pruebas que se realizan a un software específico para comprobar el buen funcionamiento de este y su calidad. El objetivo final del testing es comprobar que el software hace exactamente lo que se espera de él y para lo que se supone que fue creado.

Consta de una serie de actividades que se planean con anticipación y luego son ejecutadas como parte del proyecto. Estas actividades varían de acuerdo al tipo de proyecto que se esté realizando. Y pueden incluir métodos, herramientas y técnicas especiales de testing.

Cabe destacar que, el testing es una etapa que debe incluirse en cualquier proyecto de **IT**. Incluso en los sistemas de Business Intelligence deben realizarse estas pruebas, ya que se debe estar seguro de que lo que se va a implementar va a funcionar sin errores ni inconvenientes. Debe dársele la importancia que tiene un proyecto de **BI** en una empresa.

El testing en **BI** puede implementarse en cualquiera de sus etapas, puesto que en todas existen entradas que pueden ser testeadas. Generalmente, en la etapa de **Extract, Transform and Load (ETL)** es donde se realizan más pruebas. Sin embargo, algunas veces no se toma en cuenta la calidad de los datos y no se aplican procesos que puedan identificar la misma, como datos inválidos, no consistentes o el motivo real del error, por lo que se cataloga este como un error genérico.

Otra etapa en la que se suelen realizar pruebas es en el Front-End de la herramienta, como cuadros de mando e informes. Por lo general se realizan test manuales para comprobar aspec-

tos básicos de la información.

Las pruebas que se realizan de manera rutinaria de la estructura del sistema son para evitar que se entregue una información que no corresponda con la realidad. No obstante, se ha comprobado que esta rutina pocas veces se realiza. Un ejemplo clásico de esto son los ciclos en el Datawarehouse, los cuales causan que en dos informes que deberían contener la misma información, esta sea diferente debido a las múltiples vías existentes de resolver el caso.

En la teoría se deben implementar pruebas en todas las etapas de un sistema de BI, sin embargo, en la práctica estas pruebas suelen ser pocas y, por lo general, son instintivas contrastando los datos que se encuentran en la base de datos con los que se le muestran al usuario final. Por lo que la práctica del testing en BI debe ser una norma en todo proyecto de este tipo, para poder asegurar su correcto funcionamiento y eficacia.

5.2 Tipos de testing en Business Intelligence

Las diferentes pruebas [19] que se pueden realizar en Business Intelligence se agrupan dentro de las siguientes categorías:

- **Unit testing:** las pruebas unitarias se realizan a los componentes de una solución. Es decir, dentro de una unidad específica e individual durante el desarrollo del proyecto, no se deben realizar en las etapas posteriores, ya que lo que busca es encontrar los puntos de rotura de la solución. Los elementos que deben ser sometidos a esta evaluación son la etapa de ETL, específicamente su lógica, los datos que se extraen deben coincidir con los originales y deben tener un formato específico, por ejemplo, una fecha no puede ser de tipo texto. Igualmente, se deben probar los cálculos usados en OLAP, las reglas del negocio y la lógica de los KPI. Todas estas pruebas pueden automatizarse, ya que serán usadas a lo largo de todo el proyecto.
- **Integration testing:** la integración de pruebas se implementa si el testing unitario tuvo éxito. Se realiza para integrar dos o más módulos para poder realizarle pruebas en conjunto. Este tipo de testing busca encontrar los fallos que pudieran presentarse en la comunicación y el flujo de información entre los módulos. Es decir, asegura la construcción y despliegue con éxito, además de asegurar que no se presentarán problemas durante la ejecución del proyecto. Este tipo de testing se aplica durante el desarrollo de las bases de datos y del Datawarehouse. Implica un avance muy grande dado que asegura el buen funcionamiento del proyecto al juntar las partes que lo conforman.
- **Security testing:** esta es una prueba muy importante, no solo para los proyectos de BI

sino para todo tipo de proyecto de IT, ya que la seguridad de los datos debe ser garantizada en todo momento. Es una prueba que la realiza un especialista en ciberseguridad para evitar la irrupción de intrusos al sistema o cualquier tipo de ataque que busquen dañar o robar la información sensible y utilizarla con fines no éticos. Con estas pruebas se buscan los puntos de quiebre de la solución y se pueden implementar a tiempo las correcciones correspondientes. El pentesting o test de penetración es el más popular utilizado por los programadores.

- **Data validation testing:** se aplica al Datawarehouse para validar los datos que este contiene. La manera más común de aplicar el test es a través de una herramienta de consulta ad hoc, por ejemplo una hoja de Excel, con la que se recuperan los datos de una manera similar a los informes que arrojarán la solución. El resultado es válido cuando se encuentra un vínculo entre el informe operacional y el Datawarehouse. Siendo la persona más idónea para aplicar la prueba, un representante del negocio, que es quien tiene un mejor conocimiento de los datos y puede garantizar esta validación con éxito. Por supuesto, todo se basa en que el informe original no sea defectuoso.
- **User Acceptance testing:** estas pruebas tienen como objetivo garantizar que tanto los datos como las herramientas que se ponen a disposición cumplan con las expectativas del usuario final.
- **Performance testing:** es la prueba que mide el rendimiento de la solución. En ella se certifica que la solución va a trabajar eficientemente en condiciones reales. En este testing se deben considerar factores tales como la arquitectura del hardware, los datos, complejidad de las consultas, la rapidez de las respuestas y la escalabilidad de la solución, entre otros. Debido al tipo de performance testing que se realice, este cumplirá con una capa específica. Entre los tipos conocidos se encuentran: *stress testing*, *load testing*, *scalability testing*, *volume testing*, *endurance testing*.
- **Usability testing:** es una prueba que simula la forma en la que el usuario va a utilizar la solución. Si la misma es amigable y la forma en la que lucirá. Se mide si es accesible en todos los navegadores y los diferentes tipos de personas.
- **Regression testing:** con este test se busca probar nuevamente las funcionalidades de la solución para constatar que no han sido afectadas con el desarrollo de las bases de datos y el Datawarehouse. Cabe destacar que todos los tipos de testing que se apliquen durante el desarrollo de la solución deben quedar sujetos a regression testing.

5.3 Tipos de licencias para software de BI

Existen dos tipos de licencias [20] para el acceso a los software de BI en el mercado. Las licencias de pago, que son software propietarios y las que son gratuitas u OpenSource.

La diferencia radica en el acceso que conceden cada una de ellos. Las de software propietarios son licencias de pago que tendrán un coste específico. Algunas tienen un período de tiempo de validez para su uso comercial, otras tienen períodos de prueba con limitaciones a sus funciones, otras tienen acceso libre o un coste menor si son estudiantes, mientras que otras tienen el acceso libre solo si van a ser utilizadas para uso personal. Estas licencias por lo general incluyen soporte técnico durante la vigencia de la misma.

Las licencias OpenSource son de acceso gratuito. No incluyen soporte técnico, y algunas generan un coste por el uso de este servicio. Otras están sujetos a la ayuda en foros de la comunidad a la que pertenezca el producto.

Los software propietarios de soluciones BI tienen sus módulos integrados, tales como: OLAP, Reporting, ETL. Mientras que las licencias OpenSource las tienen por separado.

Las licencias propietarias dependen mucho del fabricante y su escalabilidad tiene procesos que suponen un coste adicional para acceder a nuevas funcionalidades. En cambio, en las licencias OpenSource se incluyen los costes de escalabilidad y los foros, blogs y comunidades pueden ayudar a resolver los inconvenientes que se presenten.

Algunas veces la seguridad del sistema es lo que diferencia a las licencias de pago de las licencias OpenSource. En los software propietarios se limitan los niveles de acceso de acuerdo a los requerimientos que tenga el cliente, mientras que a los OpenSource se tiene acceso a los informes de los que se tiene el enlace.

5.4 Herramientas de testing existentes en el mercado para BI

Actualmente, en el mercado existen diversas herramientas para realizar testing en Business Intelligence, algunas de licencia propietaria y otras OpenSource. Todas las herramientas son usadas por el testador, quien las utiliza cuando el desarrollador ha aplicado cambios en la estructura de la solución. A pesar de que algunas no son automatizables, facilitan la verificación de la calidad en cualquier entorno que se encuentre. Se destacan las siguientes:

- **Motio** [21]: es una herramienta de testing indicada especialmente para las plataformas de IBM Cognos, una de las plataformas de BI líder en el mercado. Posee tres productos: MotioPI, que es una solución OpenSource, MotioCI y MotioPI Pro las cuales son de pago.

Es excelente para la comprobación de errores en los informes y en el análisis. Admite las pruebas de regresión y de comprobación de datos cuando se han realizado cambios y los informes presentan errores de vinculación con los datos. Facilita las implementaciones, actualizaciones, control de versiones, gestión de cambios y pruebas de BI automatizadas de Cognos.

- **NBi** [22]: es una herramienta OpenSource para validar la calidad de los datos en soluciones BI. Se pueden validar todos los componentes de la solución BI elegida. Puede conectarse a cualquier base de datos ODBC u OLEDB. Además, a través de extensiones se puede conectar a una gran cantidad de bases de datos NoSQL. Y cuando se utiliza la pila de Microsoft BI, se pueden comprobar todos los componentes de las bases de datos de SQL Server y otros. Igualmente, se pueden crear cientos de casos de pruebas en minutos y hacerles el mantenimiento.
- **RightData** [23]: con esta herramienta se puede realizar la prueba de integración de datos ETL de manera de autoservicio. Garantiza la calidad de los datos como una ayuda en los procesos para el control de calidad dentro de un proyecto de BI. Es una herramienta especialmente diseñada para manejar grandes volúmenes de datos. Los usuarios pueden realizar consultas de los datos independientemente de la fuente. Tiene unas reglas prediseñadas para la comparativa de datos y, además, se pueden personalizar estas reglas.
- **QuerySurge** [24]: es una herramienta especialmente diseñada para el testing del Datawarehouse y Big Data. Permite comprobar que los datos que han sido cargados en el Datawarehouse sean correctos y tengan el formato adecuado para ser procesados. Se pueden programar las pruebas para un momento específico, ahorrando tiempo a los usuarios. Permite el testing en diferentes plataformas como *IBM, SAP, Oracle, Microsoft*, entre otros. Verifica los datos, luego los convierte y por último los actualiza durante el proceso de ETL, conectándolos desde el principio hasta el final. Ayuda a crear los escenarios de pruebas y se pueden verificar en tiempo real.
- **Datagaps BI Validator** [25]: es una herramienta con la que se pueden automatizar las pruebas de los informes de BI. Tiene una función con la que se puede comparar una consulta de la base de datos con el resultado del informe de ETL, de esta manera se puede mejorar la calidad de los datos que se muestran en el informe. Igualmente, se puede medir el rendimiento de los informes. Es muy conveniente para preparar los informes con los datos necesarios y no perder tiempo en informes que no se entienden y que usan parámetros erróneos que no aportan ningún valor al sistema. Igualmente, se pueden realizar pruebas de seguridad de BI, con las que se comprueban los accesos

a la plataforma. Además, se pueden realizar pruebas de regresión que son importantes durante todo el desarrollo del proyecto y también pruebas de estrés, donde se puede medir el rendimiento con la cantidad de usuarios de la plataforma.

5.5 Ventajas de las pruebas de BI

Realizar pruebas al proyecto de Business Intelligence conlleva una serie de ventajas [26]. El principal beneficio que se obtiene es la seguridad de que el proyecto es fiable y funcionará correctamente para la toma de decisiones críticas, en especial en ambientes donde se cuenta con una cantidad de datos enorme.

No obstante, no se debe evaluar solo un módulo del proyecto. Al contrario, se deben probar todos los módulos y cada etapa por la que pasan los datos, de esta manera se obtiene una información fiable. A continuación, se muestran algunas de estas ventajas:

- **Calidad de los datos:** cada etapa o módulo de BI que atraviesan los datos, generan una transformación. Algunas veces este proceso reduce la calidad de estos, por lo que es imprescindible que se hagan las pruebas, ya que estas garantizan la máxima calidad de los datos sin importar en qué etapa o módulo se encuentren. A su vez, permitirán que se pueda realizar un buen análisis de la información generada.
- **Reducción de riesgos:** los datos se utilizan para generar información útil para la toma de decisiones. Esto implica una gran responsabilidad, dado que muchas veces esta toma de decisiones implica procesos críticos para la empresa, por lo que si se toman decisiones basándose en datos poco fiables o que no han sido validados con anterioridad, puede provocar grandes pérdidas a la compañía. Si se le realizan pruebas a los datos en cada etapa por la que atraviesan, estos darán la información correcta, evitando riesgos innecesarios para la organización.
- **Mayor eficiencia:** los errores pueden ser detectados y corregidos en etapas iniciales, lo que ahorrará tiempo y esfuerzo que puede ser empleado en otras tareas beneficiosas para la empresa. Teniendo datos fiables, se generan informes fiables que ayudan de una forma más eficiente a la toma de decisiones.
- **Impulsa una cultura de datos:** las pruebas en los entornos BI y el análisis de los datos corrige errores, lo que provoca que se cree más confianza en una cultura de datos automatizados. Los usuarios de los entornos BI confiarán más en sus informes y les facilitará la toma de decisiones críticas.

- **Capacidad del servidor:** con las pruebas de estrés y rendimiento se pueden detectar problemas y errores antes de que sucedan en un entorno de producción. Mientras mayor sea la capacidad del servidor, mejor será el tiempo de respuesta de los informes. También se puede conocer la capacidad máxima de usuarios del entorno, de modo que pueda ser ampliada antes de generar conflictos.
- **Ambiente más activo:** las pruebas automatizadas permiten que los empleados se sientan más activos en el uso de la plataforma, dado que no deben pasar horas realizando comprobaciones y pruebas manuales que los pueden desmotivar y pueden centrarse en tareas más creativas y que aporten más valor a la compañía.
- **Historial de las pruebas:** el testing automatizado crea evidencias fiables y genera informes de los resultados, los cuales se pueden analizar para comprobar la causa de los errores o fallos que se presentan. En cualquier momento se pueden realizar las pruebas de regresión combinando otros factores que no se han comprobado anteriormente. De esta manera no se repiten pruebas que pueden arrojar el mismo resultado perdiendo tiempo en ellas.

Herramientas de Testing en BI

En este capítulo veremos una definición más completa de algunas de las herramientas de testing existentes en el mercado, así como su forma de uso, aplicación y beneficios que aportan.

6.1 TMF: Herramienta para Ruby de pruebas mínimas

TMF [27] es una herramienta minimalista cuyo objetivo es mantenerse simple, pero que funciona muy bien con Ruby. Usando solo dos métodos, `assert` y `stub`, se pueden crear pruebas muy interesantes.

Los componentes de back-end pueden ser probados con esta herramienta realizando un examen de la unidad. Es una herramienta gratuita para la cual, si requiere soporte, debe registrarse en GitHub. La última versión lanzada es la 0.1.0 del 2016.

6.1.1 Uso de TMF

Para ejemplificar el uso de TMF se muestran a continuación dos archivos que contienen una codificación específica (la cual no se muestra por lo extensa que es, pero puede ser visualizada en [Github](#)) y cómo se realiza el testing. Los archivos se encuentran en carpetas distintas y se llaman: `PROJECT_ROOT/lib/foo.rb` y `PROJECT_ROOT/test/foo_test.rb`

Los test se pueden ejecutar con la simple instrucción:

```
1 $ ruby test/foo_test.rb && echo "all tests pass"  
2 # => all tests pass
```

Listing 6.1: Ejecución de test en TMF

Y se obtendrán resultados en segundos.

6.2 Motio PI para Cognos de IBM

Motio [18] [21] posee tres versiones: **Motio PI**, **Motio CI** y **Motio PI Pro** donde la primera es gratuita y las dos últimas requieren el pago anual de las licencias para su uso. Cabe destacar que para este proyecto únicamente se explicarán las bondades de Motio PI.

Motio PI es una herramienta de testing que está diseñada para realizar pruebas de rendimiento y de regresión a la solución de BI Cognos de IBM, especialmente cuando se realizan cambios en el Datawarehouse.

Su principal ventaja es que puede ser utilizada desde cualquier ordenador o dispositivo que tenga acceso al servidor donde está alojada la solución de Business Intelligence, no necesariamente debe estar instalada en el servidor.

6.2.1 Bondades de Motio PI

Para evitar pérdida de tiempo y agilizar los procedimientos, incluida la automatización, con Motio PI se pueden lograr las siguientes tareas:

- Recuperar el contenido de usuarios Cognos que han sido eliminados, ya sea por error, fallo técnico o si han abandonado la empresa.
- También se pueden recuperar los modelos de marco eliminados, perdidos o dañados.
- Se puede consultar un contenido de Cognos específico que coincida con los criterios de búsqueda que se planteen, por ejemplo, de un paquete en particular, se pueden consultar todos los informes que pertenezcan al mismo.
- Después de validar cambios en un modelo, se puede realizar la validación de sus informes.
- Se pueden revisar el acceso, los horarios y los permisos de los usuarios de Cognos.
- Exportar a HTML.

6.2.2 Algunos test de Motio PI

Cuenta con diversos tests que pueden realizar diagnósticos sobre toda la solución BI, tales como:

- **User Content:** es un test ideal para conocer si el espacio se está utilizando de manera óptima. Se puede observar rápidamente el contenido de cada usuario, la ubicación en la que se encuentra y el espacio que está ocupando.

- **Execution Time:** es una prueba de regresión muy completa, permite conocer si una query, un informe o un análisis se está ejecutando de manera correcta o no, el tiempo de ejecución y cuando fue la última ejecución que se realizó. Se trata de una prueba muy importante dado que, tras realizar un cambio, se puede comprobar de manera tangible si este resultó correcto, si produjo mejora en el rendimiento o si, por el contrario, ocasionó algún error o empeoró su rendimiento. De esta manera, se pueden identificar los errores y corregirlos a tiempo, ya sea en el análisis, la query, el informe e incluso a nivel de datos.
- **Orphaned Objects:** es otra prueba de regresión. Esta muestra los objetos sin padre, los cuales pueden ser los causantes de los datos inconsistentes y provoquen errores en los resultados. Al poder identificarlos, se pueden dar de baja o ser corregidos.

6.3 RightData

El software RightData [23] está diseñado para SAP, siendo una excelente herramienta de pruebas para garantizar la calidad de los datos y la automatización como ayuda a los equipos tecnológicos y comerciales, entre otras ventajas. Este software tiene una licencia de pago.

El paso de **Extract, Transform and Load (ETL)** es vital en cualquier proyecto de Business Intelligence. RightData se asegura de que este proceso sea impecable con sus funciones de testing, garantizando que los datos mantengan la calidad.

Con la suite RightData se pueden realizar pruebas, validar y reconciliar los datos sin código. De esta forma, se pueden automatizar las auditorías del control de calidad de los datos y se garantiza la validez, fiabilidad e integridad de los mismos.

6.3.1 Beneficios de las pruebas de ETL con RightData

Algunos de los beneficios que obtenemos al utilizar la herramienta de RightData son los siguientes:

- Asegura que la extracción de los datos se realiza correctamente.
- Garantiza que durante la transformación de los datos los convierte al formato adecuado sin que se produzcan problemas de calidad. De esta manera se puede realizar la migración de los datos con total confianza.
- Mitiga el riesgo de errores.
- Los procesos de migración e integración se vuelven eficaces.
- Se evitan los fallos en el Datawarehouse.

- Permite manejar grandes volúmenes de datos.
- Posee un generador de escenarios muy útil para la validación de reglas comerciales.
- Capacidad de conectarse a fuentes de datos SAP y comprender los datos.
- Posibilidad de realizar comparaciones masivas entre sistemas utilizando el generador de escenarios.
- Identificación de problemas de los datos en origen antes de transformarlos y cargarlos en destino.

6.4 QuerySurge

QuerySurge [24] es una herramienta de testing ETL de Oracle, la cual funciona en una variedad de plataformas como Big Data, bases de datos tradicionales, almacenes de datos, informes de BI y archivos JSON, entre otros.

La licencia es de pago y maneja una variedad de paquetes para poder adaptar la compañía a la mejor opción. Sin embargo, posee modelos que permiten probar la herramienta de manera gratuita que van desde los tres días hasta los cuarenta y cinco días, dependiendo de la opción que se seleccione.

De acuerdo con la necesidad de la empresa, QuerySurge proporciona soluciones inteligentes automatizadas de pruebas de datos, tales como:

- Pruebas ETL y almacenamiento.
- Testing de grandes volúmenes de datos.
- Pruebas dinámicas de validación continua de datos que se pueden integrar con otras soluciones DevOps.
- Pruebas de informes de Business Intelligence donde se pueden realizar comparaciones de los datos obtenidos contra los datos originales.

6.4.1 Ventajas que ofrece QuerySurge

El uso de QuerySurge presenta las siguientes ventajas a la hora de realizar los test:

- La cobertura de validación de los datos es mucho mayor.
- Los datos críticos pueden ser optimizados a través de análisis completos.

- Compatibilidad de proyectos con una sola instancia de usuario.
- Se pueden crear consultas inteligentes sin necesidad de escribir **SQL**. También se pueden personalizar las pruebas.
- Permite visualizar las estadísticas de las pruebas en tiempo real a través de pantallas gráficas.
- Permite realizar pruebas de **BI** en Tableau, Microsoft Power BI, IBM Cognos, SAP, BusinessObjects, MicroStrategy y Oracle OBIEE.
- Puede ser instalada tanto en servidores locales como en la nube.
- Se trata de una herramienta con cifrado AES de 256 bits, por lo tanto, muy segura.
- Soporta más de doscientos almacenes de datos, ya sea como fuente o destino. Entre estas fuentes de datos se encuentran: Big Data & **NoSQL**, Bases de datos tradicionales, archivos y APIs, eCommerce, marketing y contables.

6.5 Datagaps BI Validator

Datagaps [28] permite automatizar las pruebas en plataformas **BI** tales como Microsoft Power BI, Oracle Analytics y Tableau.

Para Microsoft Power BI se puede realizar pruebas de regresión, funcionales de informes y rendimiento. Oracle Analytics admite pruebas de áreas temáticas, de paneles, de informes y de catálogos web. Mientras que Tableau Server y Online admite pruebas de Workbooks y Views.

Existen otras plataformas **BI** con soporte limitado, estas son:

- **MicroStrategy**: permite pruebas de paneles e informes.
- **SAP BusinessObjects**: admite pruebas de documentos e informes.
- **IBM Cognos**: Permite pruebas en informes de **BI**.

La tecnología de arquitectura de **ETL** que utiliza Datagaps BI Validator está patentada, lo cual garantiza una gran fiabilidad en las pruebas.

Esta herramienta tiene una licencia de pago, pero también se puede obtener una licencia gratuita de prueba por catorce días, o solicitar a Datagaps una demostración de sus productos.

6.5.1 Beneficios con Datagaps BI Validator

Existen diversos beneficios [25] que las plataformas anteriormente mencionadas pueden obtener al utilizar esta herramienta de testing, entre ellos se encuentran:

- Realización de pruebas de validación, de regresión, de estrés, de migración, de catálogo, funcionales, de rendimiento, de informes ad hoc y de colaboración empresarial.
- Permite la integración continua en BI para el desarrollo de aplicaciones futuras.
- A través de Jenkins se pueden ejecutar pruebas de manera automática.
- Los usuarios claves pueden recibir notificaciones por correo electrónico, facilitando la toma de decisiones.
- Posee una serie de informes web listos para ser usados en cualquier momento. No obstante, se pueden realizar consultas adicionales.
- Al estar integrada con la plataforma de BI, se facilita la creación de pruebas utilizando los metadatos de la misma, de esta manera no hay necesidad de realizar una programación personalizada, es decir, no se requiere de código.
- Las pruebas de BI se pueden realizar de extremo a extremo. Tiene la facilidad de poder comparar los datos con el resultado de salida de la consulta SQL.
- También se pueden comparar datos de diferentes plataformas de BI facilitando la migración de los mismos.
- También se puede realizar comparaciones de informes resumidos con datos de informes detallados.

6.6 NBi

NBi [22] es un marco OpenSource que originalmente fue creado para que funcionara con Microsoft BI, sin embargo, se extendió a otras plataformas y ahora funciona con Mondrian, MySQL y PostgreSQL. Bases de datos, cubos OLAP, bases de datos relacionales, informes y ETL's se encuentran dentro de su alcance.

Siendo un framework complemento para NUnit, su principal objetivo es permitir que los usuarios puedan realizar testing, con un enfoque declarativo, utilizando XML sin tener que acudir a códigos C# ni Visual Studio. Simplemente, se crea el archivo XML y el framework lo interpretará y reproducirá las pruebas.

6.6.1 Funcionamiento

Con NBI se pueden crear pruebas eficientes y expresivas para un proyecto de Inteligencia de Negocio. No es necesario que se creen una a una, con la solución de automatización se pueden crear conjuntos de pruebas.

GenBI de NBI es la interfaz de usuario que permitirá crear estos conjuntos de pruebas en cinco pasos:

1. Crear el conjunto de pruebas.
2. Diseñar la plantilla para las pruebas.
3. Aplicar en cada caso de prueba la plantilla creada anteriormente.
4. Definir la configuración que tendrá el conjunto de pruebas.
5. Guardar el conjunto de pruebas para ser usado en cualquier momento y las veces que sea necesario.

No solo se puede automatizar el conjunto de pruebas, sino que con NBI se puede ir mucho más lejos. Por ejemplo, con el lenguaje de macros llamado `genbiL`, se pueden ejecutar las acciones que se van a realizar en la interfaz de usuario, realizadas por el mismo `genbiL`. Es una herramienta muy poderosa que permite crear y ejecutar miles de conjuntos de pruebas en tan solo unos minutos. Igualmente, se garantiza la cohesión en el conjunto, el proceso de mantenimiento es más fácil y permite la reutilización.

En Visual Studio se puede crear rápidamente un proyecto NBI gracias al Bootstrapper que posee, con el cual se puede acelerar este proceso, creando un conjunto de pruebas en menos de un minuto sin tener que editar los archivos de configuración. Estos son ejecutados a través de la interfaz de usuario NUnit.

6.6.2 Coste de NBI

NBI es un framework OpenSource que se puede descargar libremente desde la página oficial[22]. La versión actual es la *v1.23*, que es estable.

Las soluciones de BI se pueden probar en todos sus módulos, como ETL's para validar la calidad de los datos, bases de datos, informes y cubos OLAP, entre otros.

La automatización de las pruebas se puede lograr mediante la herramienta que posee NBI, con la cual se pueden generar miles de casos en solo minutos, facilitando el mantenimiento de las mismas.

Todos los beneficios que se obtienen con NBI impactan directamente en el presupuesto de la compañía, consiguiendo ahorrar miles de euros por proyecto.

6.6.3 Ventajas del uso de NBI

Algunas de las ventajas que plantea la utilización de NBI como herramienta de testing son:

- **Codificación en XML:** Permite crear tests de forma más rápida y no depende de conocimiento específico sobre lenguajes como Java, C++...
- **Testing en múltiples componentes de los sistemas de BI:** Podemos realizar los tests a bases de datos, cubos OLAP, ETL's e informes.
- **Conexión con varios tipos de bases de datos:** Mediante el uso de la herramienta podemos testear bases de datos tradicionales como OLEDB o ODBC, pero también podemos realizar tests sobre bases de datos NoSQL.
- **Servicios Microsoft SQL Server:** Ofrece la posibilidad de testear los componentes de la base de datos de SQL Server de Microsoft, así como realizar pruebas sobre Microsoft SQL Server Integration Services (SSIS), SQL Server Analysis Services (SSAS) y SQL Server Reporting Services (SSRS).
- **Automatización de pruebas:** Nos permite crear un conjunto de pruebas y automatizarlas, de modo que resulta mucho más sencillo realizar los tests y mantener el conjunto de pruebas.

Caso de estudio: Aplicación NBI

Este capítulo está dedicado al desarrollo del caso de estudio. Contamos para ello, con la colaboración de una empresa del área textil que posee un sistema de BI completamente desarrollado sobre el cual podremos realizar la prueba de la herramienta NBI para comprobar cuánto valor aporta en el día a día en los desarrollos.

7.1 Contextualización

En este momento la compañía se encuentra en un nivel de madurez completo, *nivel siete - Formalización de Business Intelligence*. Actualmente dispone de numerosas soluciones BI que dan respuesta a problemas y preguntas rutinarias que se detectan.

En este caso concreto, el sector que nos ocupa dentro de la empresa y de las soluciones BI es el departamento *Financiero*. Al tratarse de un departamento muy delicado debido a los datos que se manejan, debemos estar seguros de que los datos son correctos y fiables y para ello debemos aplicar testing.

El departamento de *Financiero* explota los datos de los Datamarts y Datawarehouse de la compañía, los cuales son testeados por los equipos correspondientes para asegurar un dato correcto, y se usan para generar soluciones BI que aporten valor y sirvan para la toma de decisiones. En este caso, lo que nos ocupa y lo que vamos a testear con la herramienta NBI, son los informes y aplicaciones BI que diseña el equipo de *Financiero* y que son utilizados por los usuarios de negocio y también por el propio equipo de *Financiero*.

La contabilidad [29] es una de las partes más importantes de una empresa. Fue definida por el Instituto Americano de Contadores Públicos (AICPA) como *"el arte de registrar, clasificar y resumir de una manera significativa y en términos de moneda las transacciones y eventos que en su mayor parte son de carácter financiero e interpretar los resultados obtenidos"* [30]. Actualmente, la contabilidad podría definirse como *"un sistema de control y registro de los*

gastos e ingresos y otras operaciones económicas que se realizan en una organización”. Mediante la contabilidad se reportan y se justifican las actividades de la empresa a la administración pública de cada país.

Dentro del departamento de *Financiero* explotamos la información contable almacenada en los sistemas para la ayuda en la toma de decisiones, la declaración a las administraciones públicas, el control de los datos, etc.

En este caso, la organización tomó la decisión de desarrollar una aplicación propia para la generación automática de la contabilidad y adaptarla a cada nuevo país donde la empresa tuviese actividad, teniendo mayor control sobre los datos. Esta aplicación era la encargada de generar toda la contabilidad de la compañía, pero al entrar en vigor los nuevos cambios de contabilización *IFRS*¹ a nivel europeo se optó por no seguir utilizando y adaptando la aplicación y emplear una integración de otra aplicación, en este caso *SAP* [31] [32], para generar la contabilidad con las nuevas directrices.

Esto implica tener que realizar un proceso de validación entre los motores contables antiguos (*aplicación vieja*) y *SAP*, para validar y confirmar que todo se sigue generando correctamente y con los mismos tipos, de modo que terceras aplicaciones que beben del dato contable no se vean afectadas por la migración a *SAP* y se siga garantizando la compatibilidad. En definitiva, lo que se busca con estas validaciones es que la migración a *SAP* sea transparente a otras aplicaciones y que todo siga funcionando de forma precisa dado que es información crítica de la compañía.

Actualmente, ambas aplicaciones se encuentran en un período de convivencia para poder validar que la contabilidad generada por la aplicación antigua y la generada por *SAP* es la misma y que aplica los mismos tipos a los datos. Un equipo de la compañía es el encargado de realizar estas comprobaciones a mano, previo a todos los cierres contables con ayuda de unas queries *SQL* creadas específicamente para validar estos nuevos datos contra los antiguos. Lo que se pretende es automatizar estas comprobaciones de modo que no tengan que dedicar tanto tiempo a la validación los componentes del equipo y, esta sea más rápida para los cierres contables.

Para ello, se realizará una prueba con NBI con las verificaciones necesarias para el caso a validar y se comprobará si se detecta una mejora al utilizarla con respecto a la validación manual.

¹ *IFRS: International Financial Reporting Standard*: Son una serie de normas que tienen como objetivo establecer una base común para la presentación e interpretación de información financiera a nivel mundial

7.2 Desarrollo y ejecución de las pruebas

A continuación, se muestra cómo se han desarrollado las pruebas. Debido a que la información tratada de la organización es confidencial y muy sensible al tratarse de datos contables, se mostrará como valores ocultos en el trabajo.

Lo primero que se debe hacer antes de empezar a desarrollar las pruebas, es tener instalado el programa NBI en nuestro ordenador personal. Para ello, se ha elaborado una guía, adjunta en el anexo A, donde se explica cómo hacerlo.

Para realizar estas pruebas y dado que diariamente se trabaja con *Visual Studio*, se ha decidido instalar el bootstrapper en *Visual Studio 2022* como se explica en la sección A.2 para la realización de las pruebas.

7.2.1 Definición de las pruebas

Previamente, y antes de realizar la planificación a desarrollar para realizar las pruebas del caso de estudio, se han realizado numerosas pruebas con la BB.DD. gratuita *Adventure-Work2022* para poder definir correctamente los test a utilizar. Algunas de estas pruebas se han incluido en el Anexo B.

Estos test han supuesto un esfuerzo de 20h para entender el funcionamiento de la aplicación y poder disponer del conocimiento suficiente para adecuar su uso al caso que nos ocupa.

Se han realizado test para la comprobación de tipos de datos como fechas, textos y numéricos; comprobaciones de los tipos de datos de una tabla de BB.DD., comparaciones entre dos queries y comparaciones de queries con variables definidas dentro, entre otros.

Estos test han sido necesarios para comprender el funcionamiento de la herramienta. Cómo se deben configurar las cadenas de conexión, qué datos se tienen declarar para realizar las comparaciones, así como el manejo que realiza de los tipos al ejecutar las comprobaciones, entre otros. Algunos de los test realizados se pueden ver en el Anexo B y en la figura 7.1.

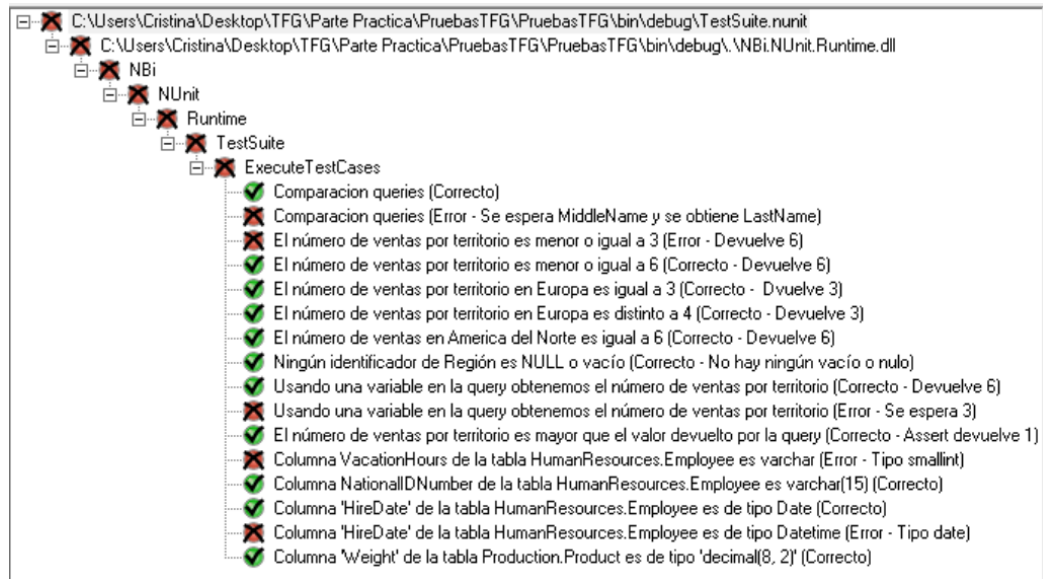


Figura 7.1: Test de prueba con AdventureWorks2022

7.2.2 Planificación y buenas prácticas

Antes de comenzar a desarrollar las pruebas, debemos establecer un plan de buenas prácticas [33] que se van a seguir para garantizar la máxima calidad.

Lo primero que debemos hacer es evaluar el trabajo que se va a realizar y crear un plan para abordarlo. Se deben planificar los objetivos que se quieren conseguir y los pasos a seguir para alcanzarlos. También se puede crear un plan de entregas en caso de tener que realizar varios tipos de pruebas. En nuestro caso no será necesario, al tratarse de una única entrega.

Una vez tenemos definidos los objetivos que se quieren alcanzar, debemos empezar a diseñar nuestros test de forma que cubra todos los casos necesarios. Una vez creados, se deben ejecutar sobre el código del informe o aplicación a validar. Con los valores arrojados por las pruebas se pueden afinar los test y mejorarlos para cubrir mejor todos los casos. Este paso se puede repetir todas las veces que sea necesario hasta tener una prueba completa y lo más fiable posible.

Por último, debemos analizar los resultados que aportan las pruebas y definir las actuaciones necesarias para mejorar el informe o aplicación con los datos generados.

Tras la ejecución del test, se puede detectar que faltan casos por probar, por lo que se tiene que volver a la fase de diseño y añadir nuevas casuísticas a probar para cubrir todos los casos pactados al inicio de la planificación.

Siguiendo estas pautas, se ha definido el plan de actuación de la siguiente forma:

- **Se establecen los objetivos a alcanzar con el test:** Se busca comprobar que ambas aplicaciones generan exactamente la misma contabilidad. Marcamos como objetivo comprobar que los tipos de los campos son iguales y los valores que generan ambas aplicaciones son idénticos.
- **Diseño del test:** Se diseña el test cubriendo todas las casuísticas posibles para cumplir los objetivos anteriormente definidos. Para ello, vamos a diseñar un test de aceptación donde se comprueba que los tipos y valores generados por ambas aplicaciones son exactamente iguales. Con esto, puede verificar si el sistema satisface los requisitos empresariales.
- **Prueba del test.** Se ejecuta el test que se ha definido para el o los casos a analizar en NBI.
- **Evaluar los resultados y obtener conclusiones.** Se comprueban los resultados generados por la aplicación NBI y se verifica si el funcionamiento de la aplicación es el esperado y se alcanzan los objetivos fijados.

La creación de un plan de actuación es necesario tanto para la validación con NBI como para la validación manual que se realiza para obtener la comparativa entre ambas.

7.2.3 Creación y ejecución de las pruebas

Con el plan ya definido, procedemos con la creación y ejecución de las pruebas del caso de estudio.

Necesitamos validar que la generación de la contabilidad de la empresa se está realizando correctamente con la nueva herramienta SAP. Para ello, tenemos desarrolladas dos queries SQL que acceden a los datos, una accede al servidor donde están alojados los datos generados por la aplicación antigua y la otra al servidor de SAP. Estas queries están acotadas por empresa y un periodo de tiempo, en este caso, el período a utilizar será de un mes para poder ser consecuentes con los cierres contables mensuales, aunque podría aplicarse para cierres trimestrales o anuales.

Con las queries se accede tanto a los datos generados por la aplicación antigua propia como a los generados por la aplicación de SAP. Al ser aplicaciones con un impacto crítico en la compañía, debemos asegurarnos de que la migración sea correcta y ambas aplicaciones devuelvan exactamente la misma información en ambos servidores.

Las queries que se van a utilizar para la prueba, son las queries con las que se realiza la validación manual por parte del equipo encargado de la empresa. Estas queries, como se ha indicado anteriormente, extraen la información cargada por ambas aplicaciones para poder

compararla. Se obtienen datos de facturas, movimientos contables, asientos contables, cuentas, importes, divisas e impuestos entre otros datos críticos.

La comprobación se ha realizado utilizando la herramienta NBI, donde se ha diseñado una prueba que nos permite comparar el resultado de ambas queries.

Como se indicó al inicio de la sección, esta prueba contiene información sensible por lo que no se puede mostrar su contenido en este trabajo por confidencialidad con el cliente, pero la estructura que se ha seguido a la hora de diseñarlo sería la siguiente:

```
1 <test name="Compare two query results">
2   <system-under-test>
3     <execution>
4       <query>
5         select * from FirstTable;
6       </query>
7     </execution>
8   </system-under-test>
9   <assert>
10    <equalTo>
11      <query>
12        select * from SecondTable;
13      </query>
14    </equalTo>
15  </assert>
16 </test>
```

Listing 7.1: Plantilla de prueba NBI

Esta es la plantilla que nos proporcionan desde la web para empezar a definir nuestra prueba de comparación. Además de definir las queries que queremos comparar, debemos definir las conexiones a las **BB.DD.**. Las cadenas de conexión pueden definirse de varias formas:

- **Definir la cadena dentro de cada test.** Podemos especificar la cadena de conexión en la etiqueta `<query>`.

```
1 <result-set>
2   <query connection-string="...">
3     ...
4   </query>
5 </result-set>
```

Listing 7.2: Definición de la cadena dentro del test

- **Definir la cadena de conexión por defecto.** Al inicio del documento en la etiqueta `<settings>` podemos definir las conexiones que se aplicaran por defecto siempre y cuando no se especifique otra dentro del test.

```
1 <settings>
2   <default apply-to="system-under-test">
3     <connection-string>...</connection-string>
4   </default>
5 </settings>
```

Listing 7.3: Definición de la cadena por defecto

- **Definir cadenas de conexión por referencia.** Al igual que en las por defecto, podemos definir varias cadenas de conexión en la etiqueta `<settings>` y asignarles un nombre mediante el cual podemos llamarlas en los test mediante el uso del símbolo `"@"`. Por ejemplo:

```
1 <settings>
2   <reference name="referencia1">
3     <connection-string>...</connection-string>
4   </reference>
5 </settings>
6 <test name="Test11">
7   <system-under-test>
8     <result-set>
9       <query connection-string="@referencia1">
10        ...
11      </query>
12    </result-set>
13  </system-under-test>
```

Listing 7.4: Definición de la cadena por referencia

- **Definir cadenas de conexión en el fichero de configuración.** Otra forma de definir las cadenas de conexión es dentro del propio fichero de configuración, donde se creará una etiqueta `<connection-strings>` donde se le asignará un nombre y la conexión a usar.

```
1 <add name="conexion1"
2   connection-string="..." />
```

Listing 7.5: Definición de las cadenas en fichero de configuración (1/2)

Para utilizar estas conexiones se definirán al inicio del documento de pruebas dentro de la etiqueta `<settings>`:

```
1 <reference name="conexion1">
2 <connection-string>@conexion1</connection-string>
3 </reference>
```

Listing 7.6: Definición de las cadenas en fichero de configuración (2/2)

Para definir las cadenas de conexión en la prueba, se ha optado por definir al inicio del documento las **conexiones por defecto** que necesitamos que se apliquen tanto en la etiqueta `<sistem-under-test>` como en la etiqueta `<assert>`. En caso de que necesitemos crear otro test con otra cadena de conexión, solo tenemos que especificar la nueva cadena en la etiqueta `<query>` del test y utilizará esa conexión en vez de la definida por defecto al inicio del documento.

En esta ocasión, se ha optado por definir las cadenas de conexión con el método por defecto, pero también podríamos haberlas definido por referencia al inicio del documento en la etiqueta `<settings>`. De esta manera, si tuviésemos más de dos cadenas de conexión distintas y estas cambiasen, no tendríamos que cambiarlas una a una, solo necesitaríamos modificar las cadenas al inicio del documento y NBI se ocuparía del resto.

Para poder comparar el resultado de dos queries, podría realizarse de varias formas según la necesidad que se presente en cada caso:

- **Definir un conjunto de resultados en un archivo externo CSV.** Podemos realizar la comparación del resultado de query contra una exportación de un fichero CSV con los resultados esperados.

```
1 <equal-to>
2 <result-set file="C:\Resultado.csv" />
3 </equal-to>
```

Listing 7.7: Definición del conjunto de resultados en un archivo externo

- **Definir un conjunto de resultados en el test.** Se pueden definir las filas de salida esperadas y el test comparará el resultado de la query contra lo que hayamos definido en el test como resultado deseado.

```
1 <equal-to>
2 <result-set>
3 <row>
4 <cell>Primer valor esperado</cell>
```



```
5         <cell>100.05</cell>
6     </row>
7 </result-set>
8 </equal-to>
```

Listing 7.8: Definición del conjunto de resultados en el test

- **Definir un conjunto de resultados en un archivo XML.** Podemos definir un conjunto de resultados en un archivo XML y comparar la query directamente contra él.

```
1 <equal-to>
2   <xml-source>
3     <file>MyFile.xml</file>
4     <xpath>
5       <from>//Path</from>
6       <select attribute="Id">//Path/Item</select>
7       <select>//Path/Item/SubItem</select>
8     </xpath>
9   </xml-source>
10 </equal-to>
```

Listing 7.9: Definición del conjunto de resultados en un archivo XML

- **Comparar directamente contra otra query.** Realizamos la comparación ejecutando ambas queries a la vez y comparando ambos resultados. De esta forma garantizamos que no ha habido cambios en ninguna de las BB.DD. en el momento de realizar la prueba que puedan interferir en el resultado.

```
1 <equal-to>
2   <query connection-string="...">
3     SELECT * FROM Tabla1
4   </query>
5 </equal-to>
```

Listing 7.10: Definición de la comparación directa entre queries

Para realizar esta prueba, se ha optado por **comparar ambas queries directamente** evitando así cambios en las BB.DD. que pueden interferir en los resultados de la prueba.

Para el análisis de la queries y la definición de los requisitos se han necesitado **6 horas**. El análisis y la definición son compartidas en ambas validaciones, por lo que este tiempo no se tendrá en cuenta en la comparativa entre ambas validaciones.

El diseño del test ha requerido de **2 horas** y no se han tenido en cuenta para esta cifra el tiempo invertido en el aprendizaje de la herramienta, ni el invertido en la creación del plan de pruebas.

El test resultante con los datos confidenciales enmascarados sería de la siguiente forma:

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <testSuite name="Test validacion migracion a SAP"
3   xmlns="http://NBI/TestSuite">
4   <settings>
5     <default apply-to="system-under-test">
6       <connectionString>conexion-SAP</connectionString>
7     </default>
8     <default apply-to="assert">
9       <connectionString>conexion-DB2</connectionString>
10    </default>
11  </settings>
12  <test name="Comparacion de resultados de las dos queries">
13    <system-under-test>
14      <execution>
15        <query>
16          Query_SAP;
17        </query>
18      </execution>
19    </system-under-test>
20    <assert>
21      <equalTo>
22        <column index="0" role="key" type="numeric"/>
23        <column index="1" role="value" type="text"/>
24        <column index="2" role="value" type="boolean"/>
25        <column index="3" role="value" type="dateTime"/>
26        <...>
27      </equalTo>
28      <query>
29        Query_DB2;
30      </query>
31    </assert>
32  </test>
</testSuite>
```

Listing 7.11: Caso de estudio. Test con NBI

Una vez creado el test, se ejecuta con NUnit para comprobar si se está generando la contabilidad correctamente. Al lanzar el programa mediante *F5*, abre una nueva pestaña y ejecuta los test que tenemos definidos. En este caso solo necesitamos definir un único test donde com-

para ambas queries y obtener su resultado.

Si el test fallase, mostraría una pantalla como la de la figura 7.2, en donde se explican los datos mostrados. También se puede exportar el resultado del test a un documento XML desde la pestaña *Tools\Save Results as XML* para ver el resultado del test más en detalle.

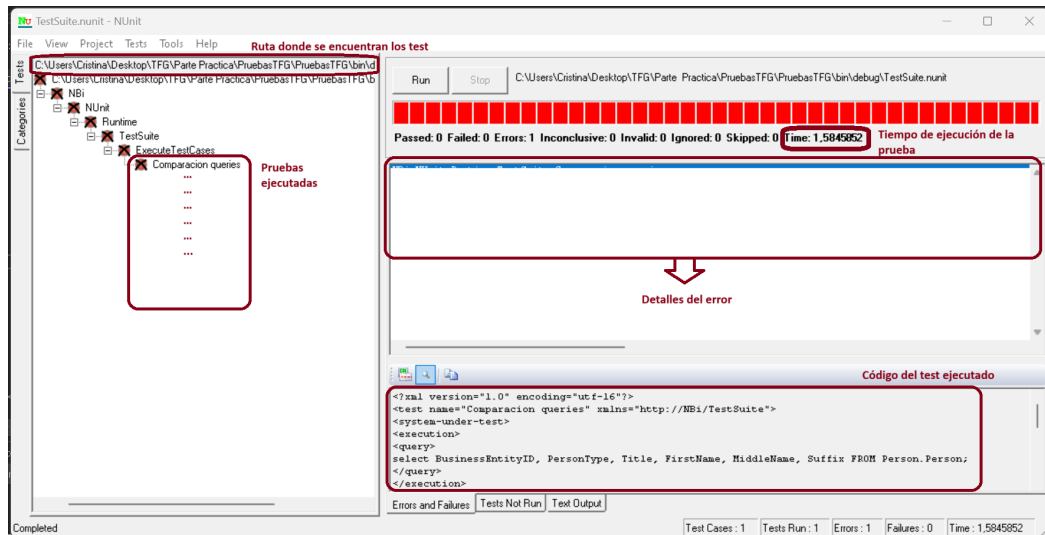


Figura 7.2: Caso de estudio. Test erróneo

En la figura 7.3, se puede ver el detalle del error que devuelve el test. En él, se muestra el error detectado, el número de filas obtenidas, el resultado esperado y el resultado obtenido.

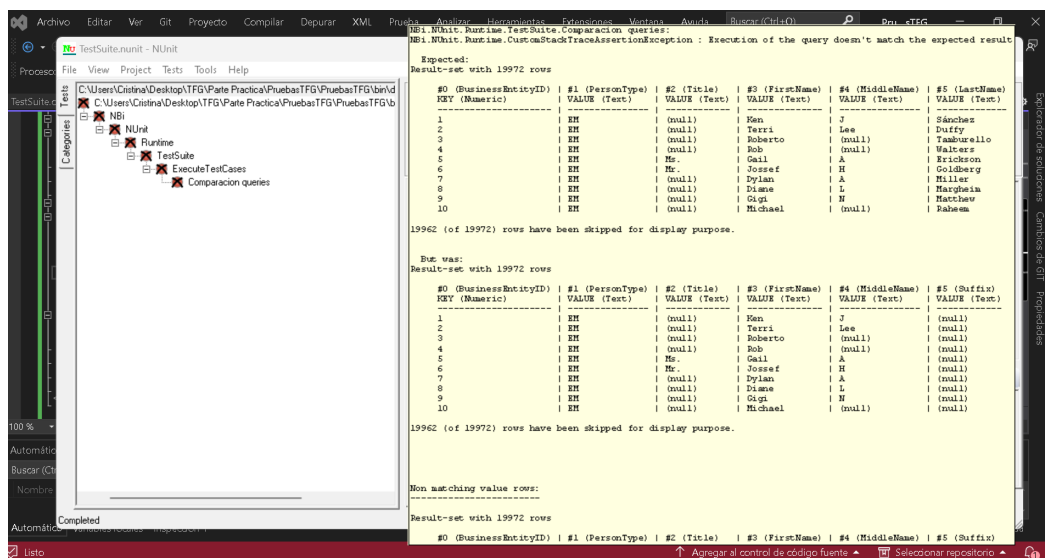


Figura 7.3: Caso de estudio. Explicación error

Si el resultado es correcto, la pantalla que mostrará será la de la figura 7.4. Indicará el tiempo de ejecución de la prueba y que su resultado ha sido correcto.

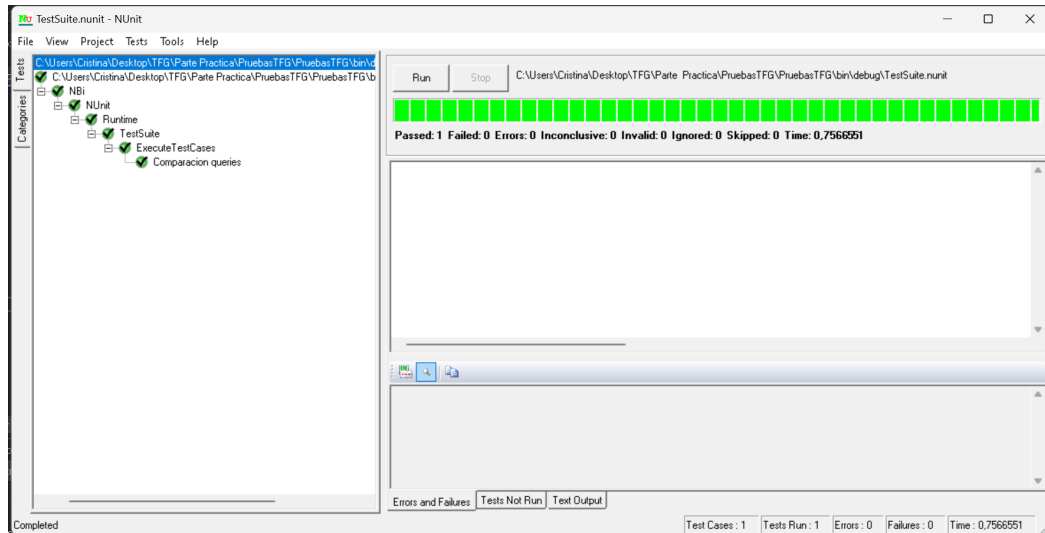


Figura 7.4: Caso de estudio. Ejecución correcta del test

Nota: Las figuras mostradas en este capítulo del trabajo han sido generadas con test de prueba creados con la BB.DD. de Microsoft SQL Server AdventureWorks2022, debido a la confidencialidad de los datos del cliente.

El test realizado con los datos del cliente para una empresa y un mes natural ha tardado **9 minutos 15 segundos** en ejecutarse y finalizar correctamente. Para realizar la prueba completa, el test debería lanzarse para todas las empresas de la compañía para el mes a validar.

7.3 Comparativa y conclusiones

En esta sección se realiza una comparativa sobre el tiempo que le tomaría a un empleado realizar las validaciones a mano y el tiempo que se necesita para realizar las pruebas con la herramienta NBI y compararlos.

7.3.1 Validación manual

Actualmente, la validación manual de la generación de la contabilidad la realiza un equipo de la propia compañía. El método que siguen para la validación entre ambas aplicaciones es generar las exportaciones de ambas queries para una empresa y un período de tiempo, normalmente un mes, y compararlas.

Para realizar la validación, los resultados se exportan en archivos *CSV* que luego se cargan en un Excel para efectuar las modificaciones y los cruces necesarios entre ambos usando la tecnología de *Power Query* para poder ejecutar correctamente la validación.

Una vez cargados todos los datos en el fichero Excel, para realizar las comprobaciones, se deben crear tablas dinámicas por ciertos campos clave, por ejemplo por asientos contables, y comprobar el número de movimientos que existen en cada asiento para ambas aplicaciones como se muestra en la figura 7.5.

Etiquetas de fila	Suma IMPORTE	Suma de MOVOS. N°	Etiquetas de fila	Suma IMPORTE	Suma de MOVOS. N°
7000166		1	7000166		1
7000167		3	7000167		3
7010004		7	7010004		7
7010005		1	7010005		1
7010007		1	7010007		1
7010011		9	7010011		9
7010012		3	7010012		3
7010013		2	7010013		2
7010014		5	7010014		5
7010015		1	7010015		1
7010016		1	7010016		1
7010017		1	7010017		1
7010018		3	7010018		3
7010019		2	7010019		2

Figura 7.5: Validación. Comprobación con tablas dinámicas

También se realiza una comparación entre las exportaciones Excel de los resultados de las queries con ayuda de una herramienta de comparación de la que dispone la compañía, en este caso *Spreadsheet Compare*, donde permite comprobar que los tipos coinciden, los valores son iguales o si se han añadido o eliminado filas. En la figura 7.6 podemos ver un ejemplo de comparativa donde las filas entre ambos Excel no son las mismas.

Figura 7.6: Validación. Comparación de las exportaciones Excel

Todo esto se debe hacer para cada empresa y periodo de tiempo que se necesite reportar. Todas estas comprobaciones, son necesarias para tratar de garantizar la existencia de los mis-

mos datos en ambas aplicaciones. Realizar estas comprobaciones a mano es muy laborioso y se necesita mucho tiempo.

El tiempo para ejecutar ambas queries ocupa **9 minutos 15 segundos**. La carga de ambas exportaciones en el fichero Excel depende del tamaño de estas, en la prueba que se ha realizado con una única empresa se han necesitado **45 minutos** para realizar la carga y las transformaciones necesarias con *Power Query*. Además, para la creación de las tablas dinámicas y las comprobaciones con estas, se han necesitado **2 horas 30 minutos**.

También se debe tener en cuenta el tiempo empleado en la comparación de los ficheros Excel con la herramienta de comparación. Ha tomado **30 minutos** realizar estas comprobaciones.

7.3.2 Conclusiones sobre la validación

Se va a realizar una comparativa entre los tiempos empleados para realizar la prueba con NBI y los empleados en las validaciones manuales. Estos se reflejan en la tabla 7.1.

Acción	NBi	Manual
<i>Creación de la prueba</i>	2 h	-
<i>Ejecución</i>	9 min 15 s	9 min 15 s
<i>Carga de datos en Excel</i>	-	30 min
<i>Análisis</i>	5 min	2 h 30 min
<i>Comparación de Extracciones</i>	-	30 min
Total	2 h 14 min 15 s	3 horas 39 min 15 s

Tabla 7.1: Tiempos de validación.

Como se puede observar en la tabla anterior 7.1, hemos necesitado **2 horas 14 minutos y 15 segundos** para realizar la prueba de validación con la herramienta NBI, mientras que para realizar la validación de forma manual hemos precisado **3 horas 39 minutos y 15 segundos**. Comparando estos tiempos ya se puede ver una clara mejora y, si la cuantificamos, **esta representa una mejora del 38,8%** en los tiempos de validación.

Pero realmente la mejora es aún mayor dado que, como se indicó al principio de las pruebas, el estudio realizado es sobre una única empresa y un mes natural. La validación debe realizarse para todas las empresas de la compañía para el cierre contable, por lo que el tiempo

empleado para la validación manual se repetirá para cada una de las empresas al tener que volver a realizar siempre los mismos pasos. Mientras que con la validación con NBI, el tiempo que se necesitará será menor.

Para realizar las pruebas con NBI tenemos dos opciones:

- **Modificar el test.** Podríamos modificar en el test que ya hemos definido las veces que sea necesario las query de validación para comprobar todas las empresas.
- **Replicar el test.** Se podría replicar el test para cada una de las empresas y validar todo junto como en el ejemplo que se muestra a continuación:

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <testSuite name="Test validacion migracion a SAP"
3  xmlns="http://NBI/TestSuite">
4      <settings>
5          <default apply-to="system-under-test">
6              <connectionString>conexion-SAP</connectionString>
7          </default>
8          <default apply-to="assert">
9              <connectionString>conexion-DB2</connectionString>
10         </default>
11     </settings>
12     <test name="Comparacion empresa 1">
13         <system-under-test>
14             <execution>
15                 <query>
16                     Query_SAP;
17                 </query>
18             </execution>
19         </system-under-test>
20         <assert>
21             <equalTo>
22                 <column index="0" role="key" type="numeric"/>
23                 <column index="1" role="value" type="text"/>
24                 <column index="2" role="value" type="boolean"/>
25                 <column index="3" role="value" type="dateTime"/>
26                 <...>
27             <query>
28                 Query_DB2;
29             </query>
30         </equalTo>
31     </assert>
32 </test>
```

```
32     <test name="Comparacion empresa 2">
33         <system-under-test>
34             <execution>
35                 <query>
36                     Query_SAP;
37                 </query>
38             </execution>
39         </system-under-test>
40         <assert>
41             <equalTo>
42                 <column index="0" role="key" type="numeric"/>
43                 <column index="1" role="value" type="text"/>
44                 <column index="2" role="value" type="boolean"/>
45                 <column index="3" role="value" type="dateTime"/>
46                 <...>
47             <query>
48                 Query_DB2;
49             </query>
50         </equalTo>
51     </assert>
52 </test>
53 <test>
54     "test 3"
55 </test>
56 <test>
57     "test 4"
58 </test>
59     ...
60 </testSuite>
```

Listing 7.12: Caso de estudio. Réplica del test en NBI para cada empresa

Dado que la prueba se va a realizar todos los meses, lo más recomendable es replicar el test para cada una de las empresas una única vez y guardarlo, de modo que el siguiente mes solamente se tenga que ejecutar sin necesidad de realizar ningún tipo de modificación.

Esta validación, usando la herramienta NBI, supone un gran ahorro de tiempo y dinero a largo plazo para la compañía, aunque ya se pueden ver sus beneficios desde el primer uso.

Conclusiones y trabajos futuros

La Inteligencia de Negocio, Inteligencia Empresarial o Business Intelligence (BI) ayuda a las empresas y organizaciones en la toma de decisiones periódicas y críticas, permite que se pueden resolver problemas que son urgentes, además de tener realmente una visión del negocio en el momento, entre muchos otros muchos beneficios.

El manejo de los datos dentro de una empresa cada día es más complicado por la cantidad de datos que se recopilan en diferentes sistemas dentro de una misma organización. Cada área funcional tiene su propia forma de manejar los datos, por lo que se hace complicado al momento de organizar toda la información. Es por esta razón que las herramientas de Business Intelligence se han convertido en un sistema muy provechoso dentro de cualquier compañía, ya que un proyecto de Business Intelligence se puede aplicar en cualquier tipo de empresa.

Implementar un proyecto de BI no es una tarea inmediata que pueda ser realizada en unos pocos días, necesita tiempo para adaptarlo correctamente a la empresa y a las diferentes áreas funcionales que se incluyan. Por lo que aplicar el testing a los sistemas de BI es muy importante en el momento de implementarlos.

Un sistema de BI recopila datos, los procesa y luego realiza análisis con ellos para obtener resultados que sirvan en la toma de decisiones. Cada etapa del BI debe ser probada de alguna manera para asegurarse de que el mismo funciona perfectamente y que la información que generará va a ser fiable.

Para lograr esto, las herramientas de Testing de BI aportan todas las funcionalidades que permitan realizar pruebas durante la extracción de los datos y transformarlos en información útil y de gran valor para la empresa. Para este proyecto, se ha utilizado la herramienta NBI, la cual es una excelente solución OpenSource para el testing de sistemas de BI, ya que con ella se pueden probar todos los módulos del sistema como los ETL's, informes, bases de datos, cubos OLAP, etc.

8.1 Conclusiones

Como se ha podido comprobar a lo largo del trabajo, el uso de los sistemas de BI aporta grandes beneficios a la compañía si se implementa correctamente y se le hacen las preguntas adecuadas. Aunque los sistemas BI por sí solos no bastan para asegurar una mejora en la compañía, es de vital importancia aplicar correctamente el testing en cada una de las fases del desarrollo de los proyectos BI. De este modo, se puede garantizar la calidad de los datos que se van a usar y que estos son fiables. Es importante realizar una buena planificación de las pruebas que se deben realizar para poder garantizar la calidad y fiabilidad de la información.

El beneficio que aporta el uso de las herramientas de testing BI se puede medir en tiempo y costo para la compañía, sobre todo si se aplican a largo plazo. Además, ayuda a garantizar de manera más eficaz la calidad y fiabilidad de los datos que se utilizan para la toma de decisiones. Asimismo, ha quedado demostrado que el uso de las herramientas testing no solo reporta beneficio a la empresa, sino que también lo hace al empleado; el cual, será más rápido a la hora de realizar las validaciones de sus desarrollos, lo que le proporcionará más tiempo para desempeñar otras tareas más creativas que requieran de su atención y no tener que utilizar su tiempo en tareas monótonas y repetitivas que no le suele gustar desempeñar.

Los objetivos alcanzados tras la realización de este proyecto son:

- Mayor dominio sobre los cambios que se realizan en los proyectos. Al aplicar testing, tenemos controlados los cambios que se realizan y podemos ver si suponen algún cambio o producen algún impacto en el resultado esperado del proyecto y actuar antes de que se transformen en un problema.
- Mediante la implementación y uso del testing en proyectos BI, reducimos el tiempo empleado por un trabajador en la validación de los cambios. Como se pudo ver durante el desarrollo del proyecto, desde el primer uso se obtiene una mejora en los tiempos, lo cual proporciona al trabajador más tiempo para poder desempeñar tareas más dinámicas y menos repetitivas.
- Con la aplicación del testing también garantizamos la calidad de los datos que, a su vez, nos ayuda a mejorar y garantizar la calidad de los desarrollos BI.

Para el caso particular de la solución testing implementada en el caso de estudio de la empresa, es posible apreciar cómo el uso de NBI puede ayudar a generar una mejora del 38,8% del tiempo empleado con respecto a una validación realizada de forma tradicional sobre el desarrollo del área financiera de la empresa. Esta mejora supone un gran ahorro para la compañía, tanto de tiempo como de esfuerzo.

En conclusión, el uso de las herramientas de testing nos permiten hallar más rápido los posibles errores que se puedan producir en los desarrollos, generando un mayor control de los impactos sobre los proyectos y originando mayor confianza en los datos al mejorar su calidad. Por tanto, aporta un gran beneficio de tiempo y esfuerzo en las tareas diarias de los empleados como para la compañía.

8.2 Trabajos futuros

Como trabajo a futuro con la herramienta y una vez tenemos desarrollados los test para los cierres contables tras este proyecto, se plantea automatizar el conjunto de pruebas, de modo que se automatice el proceso de validación y revisión de los datos todos los meses sin necesidad de depender de todo un equipo para su validación.

También se plantea seguir ampliando el uso de la herramienta en otro tipo de desarrollos, aplicándola a desarrollos de informes BI y creando otros conjuntos de pruebas que permitan agilizar los test para varios casos recurrentes que se prueban en el día a día en los desarrollos y modificaciones de otros proyectos.

Además, se podría usar para implementar test para controlar las ETL's dependientes del equipo de desarrollo de BI y los cuadros de mando de Microsoft Power BI para agilizar las validaciones de los cambios.

Apéndices

Guía de instalación

En este apartado se ha elaborado una pequeña guía de instalación utilizando los recursos proporcionados por la propia web de la aplicación. Existen dos modos de instalar NBI en nuestro equipo:

- **Instalación manual.**
- **Instalación mediante Visual Studio.**

A.1 Instalación manual

Para configurar NBI en un equipo, se deben cumplir una serie de requisitos previos:

- *.Net Framework 4.5 o superior.* Este puede ser descargado directamente del centro de descargas de [Microsoft](#).
- *Última versión de UNit 2.x.* La cual se consigue en la página oficial de [NUnit.org](#).
- *AdomdClient versión 1.10.0.19 y posteriores.* Si se cuenta con esta versión, ya se tienen los archivos binarios de *Microsoft Analysis Services Adomd Client*, originalmente estos vienen con Microsoft SQL Server 14, por lo que no es necesario que se haga una instalación manual.
- *AdomdClient versiones anteriores a 1.10.0.19.* Para las versiones anteriores se necesitará descargar *Microsoft Analysis Services Adomd Client* por separado desde la página oficial de MSDN.

Si se van a realizar consultas a un cubo [OLAP](#) es necesario el componente *Adomd.net*. Durante la instalación de NBI se buscará la versión 10 la cual funciona con SQL 2012, pero si necesita la versión 11, no existe inconveniente en instalarla, ya que NBI puede funcionar con

múltiples versiones de *Adomd.net* en el mismo ordenador o servidor. Al haber cubierto todos los requisitos previos, se procede con la descarga e instalación de la última versión de NBI que podemos encontrar disponible en [GitHub](#) [22].

También debemos tener en cuenta la configuración del entorno de pruebas. En cualquier proyecto de IT debe considerarse este aspecto. El entorno de pruebas debe ser igual al de producción para que pueda funcionar correctamente.

Los entornos de pruebas por lo general contienen la siguiente jerarquía de directorios: Un directorio base, por ejemplo, *MyTests*, y dentro se incluirán las siguientes carpetas como se muestra en la figura A.1:

- Una carpeta destinada al marco NBI.
- Tres carpetas para elementos de NBI que obtenemos al descargar el zip con la herramienta, tales como *Samples*, *Framework* y *Genbi*.
- Una carpeta destinada a cada uno de los proyectos de prueba que se vayan a realizar, en este caso una, que se denominará *MyTestsProject*.

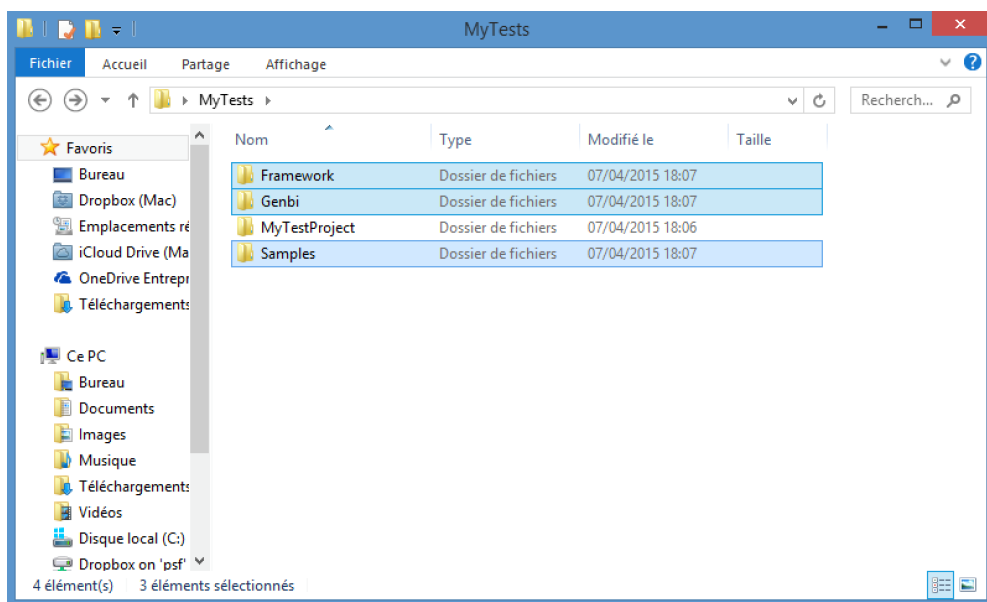


Figura A.1: Instalación NBI. Organización directorio base

Una vez dentro de la carpeta *MyTestsProject*, se crean tres archivos de texto con una sintaxis XML, los cuales pueden ser creados y leídos por *Notepad*, *Notepad++*, *Visual Studio* o por cualquier otro editor de texto. También pueden ser creados por el editor de proyectos de NBI. Estos archivos son los siguientes:

1. **Nombre_archivo.config**: este archivo hace referencia al conjunto de pruebas y es bastante corto de texto. En él se define la ruta que va a tomar el conjunto de pruebas, el tipo de NBi que se trabajará y la sección del elemento XML.

```

1      <?xml version="1.0" encoding="utf-8" ?>
2      <configuration>
3          <configSections>
4              <section name="nbi" type="NBi.NUnit.Runtime.NBiSection,
NBi.NUnit.Runtime"/>
5          </configSections>
6          <nbi testSuite="MyTestProject\MyTestProject.nbits"/>
7      </configuration>

```

Listing A.1: Configuración del archivo .config

2. **Nombre_archivo.nunit**: en este archivo se hace referencia al archivo de configuración y del framework de NBi (.dll). Además, contiene datos muy importantes como: la ruta al archivo de configuración, la ruta de ensamblaje que se refiere al marco NBi y el directorio raíz o base de aplicaciones. Se puede crear utilizando el editor de proyectos NUnit, se puede ver el paso a paso en la web de NBi.io o mediante un archivo de texto con el siguiente código:

```

1      <NUnitProject>
2          <Settings activeconfig="Default" processModel="Default"
domainUsage="Default" />
3          <Config
4              name="Default"
5              binpathtype="Auto" appbase="..\\"
6              configfile="MyTestProject\MyTestProject.config">
7              <assembly path="Framework\NBi.NUnit.Runtime.dll" />
8          </Config>
9      </NUnitProject>

```

Listing A.2: Configuración del archivo .nunit

3. **Nombre_archivo.nbits**: este es el archivo de pruebas real, en él es donde se van a crear los test. Por ejemplo, como en los casos de prueba, se puede definir una prueba que teste que una consulta es igual a otra consulta:

```

1      <?xml version="1.0" encoding="utf-8"?>
2      <testSuite name="My first test suite"
xmlns="http://NBi/TestSuite">

```

```
3      <test name="My first test">
4          <system-under-test>
5              <result-set>
6                  <query connection-string="...">
7                      <![CDATA[
8                          SELECT
9                              {[Measure].[MyMeasure]} ON 0,
10                             {[MyDimension].[MyHierarchy].Members} ON 1
11                             FROM MyCube
12                             ]]>
13                  </query>
14              </result-set>
15          </system-under-test>
16          <assert>
17              <equal-to>
18                  <query connection-string="...">
19                      SELECT MyHierarchy, MyMeasure FROM MyTable
20                  </query>
21              </equal-to>
22          </assert>
23      </test>
24  </testSuite>
```

Listing A.3: Configuración del archivo .nbts

Después de haber creado los archivos de pruebas sólo queda ejecutarlos. Debemos abrir la interfaz de usuario de NUNit, y buscar y abrir el proyecto. NUnit lo leerá y el usuario podrá elegir cuáles son las pruebas a realizar. Por último, solo debemos darle a ejecutar y NBI se encargará de realizar las pruebas y reportar los fallos que detecte.

A.2 Instalación mediante Visual Studio

En cambio, si lo que queremos es no tener que crear ningún archivo de configuración y tenemos instalado en nuestro equipo *Visual Studio*, lo que vamos a hacer es instalar el Bootstrapper creado para él.

Lo primero que debemos hacer es iniciar el programa *Visual Studio*, en este caso la versión 2022, y crear un nuevo proyecto (*Archivo\Nuevo\Proyecto...*). Este proyecto debe ser de tipo: *Class Library (.NET Framework)* o *Biblioteca de clases(.Net Framework)*, como se muestra en la figura A.2.

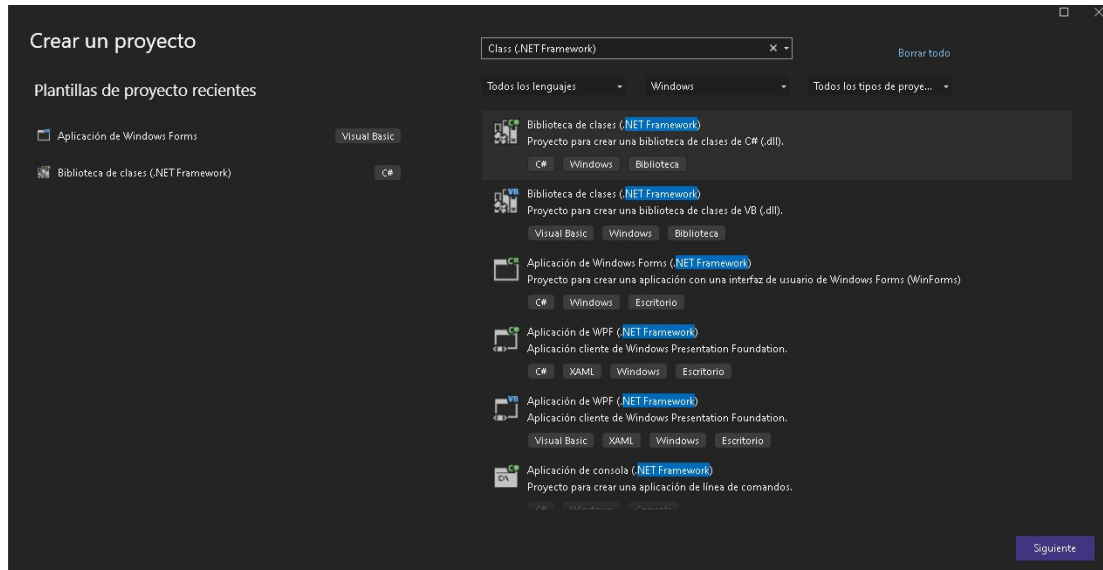


Figura A.2: Instalación NBI. Creación del proyecto en Visual Studio 2022

Una vez seleccionado, le damos el nombre y seleccionamos la ruta en la que queremos que se almacene y le damos a guardar. Una vez creado el proyecto, debemos abrir la "Consola del administrador de paquetes" para instalar NBI. Si no nos aparece por defecto, debemos buscar en *Ver\Otras ventanas\Consola del administrador de paquetes* como se indica en la figura A.3.

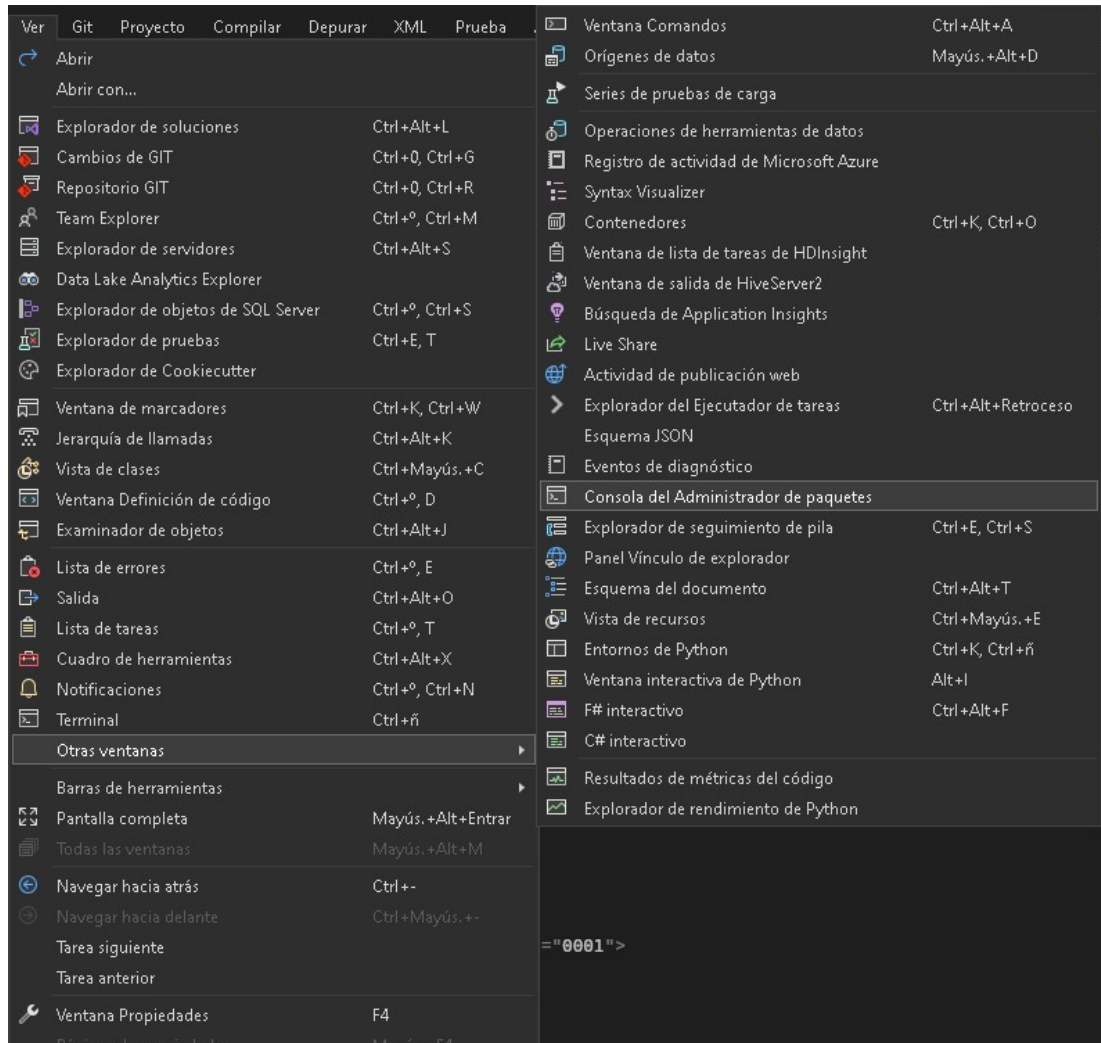


Figura A.3: Instalación NBI. Consola del administrador de paquetes

Quando tengamos visible la consola, procedemos a instalar NBI. Para ello debemos introducir el comando:

```
1 Install-Package NBI.VisualStudio
```

Listing A.4: Comando de instalación de NBI en Visual Studio

El programa se encarga de descargar automáticamente las últimas versiones tanto de NUnit como de NBI. Adaptará la solución y realizará la creación y configuración de todos los ficheros necesarios (*.nbits*, *.config*, *.nunitproj*), por lo que el usuario solo necesita crear su conjunto de pruebas en el archivo *.nbitsy* ejecutarlo mediante F5 una vez haya finalizado; de esta manera se acelera el proceso de instalación.

Test de prueba básicos con NBI

Los test que se muestran a continuación, son algunos de los test básicos que se han creado para comprender el funcionamiento y manejo de tipos que realiza la herramienta NBI.

Como se indicó en la subsección 7.2.1, estos test han sido necesarios para comprender cómo se deben configurar las cadenas de conexión, qué datos son necesarios para la ejecución de las comparaciones entre queries SQL, cómo crear variables y cómo se usan en las comparaciones, qué tipos de comparaciones podemos aplicar en las pruebas, etc. En general, el funcionamiento de la aplicación al ejecutar las pruebas. Estos test deben crearse en el fichero *.nbits*, y son los siguientes:

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <testSuite name="Test de aceptación" xmlns="http://NBI/TestSuite">
3   <settings>
4     <default apply-to="system-under-test">
5       <connectionString>Data
6       Source=LAPTOP-QTV6LEEH\SQLEXPRESS;Initial
7       Catalog=AdventureWorks2022;User
8       Id=sa;password=Pass1234</connectionString>
9     </default>
10    <default apply-to="assert">
11      <connectionString>Data
12      Source=LAPTOP-QTV6LEEH\SQLEXPRESS;Initial
13      Catalog=AdventureWorks2022;User
14      Id=sa;password=Pass1234</connectionString>
15    </default>
16  </settings>
17  <variables>
18    <variable name="NumeroSeis">
19      <query-scalar
20        connectionString="Data
```

```

15     Source=LAPTOP-QTV6LEEH\SQLEXPRESS;Initial
16     Catalog=AdventureWorks2022;User Id=sa;password=Pass1234"
17     >
18     select 6
19     </query-scalar>
20     </variable>
21 </variables>
22 <test name="Comparacion queries (Correcto)">
23     <system-under-test>
24     <execution>
25     <query>
26         select BusinessEntityID, PersonType, Title, FirstName,
27         MiddleName, LastName FROM Person.Person;
28     </query>
29     </execution>
30 </system-under-test>
31 <assert>
32     <equalTo>
33     <column index="0" role="key" type="numeric"/>
34     <column index="1" role="value" type="text"/>
35     <column index="2" role="value" type="text"/>
36     <column index="3" role="value" type="text"/>
37     <column index="4" role="value" type="text"/>
38     <column index="5" role="value" type="text"/>
39     <query>
40         select BusinessEntityID, PersonType, Title, FirstName,
41         MiddleName, LastName FROM Person.Person;
42     </query>
43     </equalTo>
44 </assert>
45 </test>
46 <test name="Comparacion queries (Error - Se espera MiddleName y
47 se obtiene LastName)">
48     <system-under-test>
49     <execution>
50     <query>
51         select BusinessEntityID, PersonType, Title, FirstName,
52         LastName FROM Person.Person;
53     </query>
54     </execution>
55 </system-under-test>
56 <assert>
57     <equalTo>
58     <column index="0" role="key" type="numeric"/>
59     <column index="1" role="value" type="text"/>
60     <column index="2" role="value" type="text"/>

```

```

55     <column index="3" role="value" type="text"/>
56     <column index="4" role="value" type="text"/>
57     <query>
58         select BusinessEntityID, PersonType, Title, FirstName,
59         MiddleName FROM Person.Person;
60     </query>
61     </equalTo>
62 </assert>
63 </test>
64 <test name="El número de ventas por territorio es menor o igual a
65 3 (Error - Devuelve 6)">
66     <system-under-test>
67         <result-set>
68             <query>
69                 select
70                 [Name], [CountryRegionCode]
71                 from
72                 [Sales].[SalesTerritory]
73                 where
74                 [Group]='North America'
75             </query>
76         </result-set>
77     </system-under-test>
78     <assert>
79         <row-count>
80             <less-than or-equal="true">3</less-than>
81         </row-count>
82     </assert>
83 </test>
84 <test name="El número de ventas por territorio es menor o igual a
85 6 (Correcto - Devuelve 6)">
86     <system-under-test>
87         <result-set>
88             <query>
89                 select
90                 [Name], [CountryRegionCode]
91                 from
92                 [Sales].[SalesTerritory]
93                 where
94                 [Group]='North America'
95             </query>
96         </result-set>
97     </system-under-test>
98     <assert>
99         <row-count>
100             <less-than or-equal="true">6</less-than>

```

```

98     </row-count>
99     </assert>
100  </test>
101  <test name="El número de ventas por territorio en Europa es igual
102  a 3 (Correcto - Devuelve 3)">
103    <system-under-test>
104      <result-set>
105        <query>
106          select
107            [Name], [CountryRegionCode]
108          from
109            [Sales].[SalesTerritory]
110          where
111            [Group]='Europe'
112        </query>
113      </result-set>
114    </system-under-test>
115    <assert>
116      <row-count>
117        <equal>3</equal>
118      </row-count>
119    </assert>
120  </test>
121  <test name="El número de ventas por territorio en Europa es
122  distinto a 4 (Correcto - Devuelve 3)">
123    <system-under-test>
124      <result-set>
125        <query>
126          select
127            [Name], [CountryRegionCode]
128          from
129            [Sales].[SalesTerritory]
130          where
131            [Group]='Europe'
132        </query>
133      </result-set>
134    </system-under-test>
135    <assert>
136      <row-count not="true">
137        <equal>4</equal>
138      </row-count>
139    </assert>
140  </test>
141  <test name="El número de ventas en America del Norte es igual a 6
142  (Correcto - Devuelve 6)">
143    <system-under-test>

```

```

141     <result-set>
142     <query>
143         select
144             [Name], [CountryRegionCode], [Group]
145         from
146             [Sales].[SalesTerritory]
147     </query>
148 </result-set>
149 </system-under-test>
150 <assert>
151     <row-count>
152     <filter>
153         <alias column-index="2">Region</alias>
154         <predicate name="Region" type="text">
155             <equal>North America</equal>
156         </predicate>
157     </filter>
158     <equal>6</equal>
159 </row-count>
160 </assert>
161 </test>
162 <test name="Ningún identificador de Región es NULL o vacío
163 (Correcto - No hay ningún vacío o nulo)">
164 <system-under-test>
165     <result-set>
166     <query>
167         select
168             [Name], [CountryRegionCode], [Group]
169         from
170             [Sales].[SalesTerritory]
171     </query>
172 </result-set>
173 </system-under-test>
174 <assert>
175     <no-rows>
176     <alias column-index="1">Code</alias>
177     <predicate name="Code" type="text">
178         <empty or-null="true"/>
179     </predicate>
180 </no-rows>
181 </assert>
182 </test>
183 <test name="Usando una variable en la query obtenemos el número
184 de ventas por territorio (Correcto - Devuelve 6)">
    <system-under-test>
        <result-set>

```

```

185     <query>
186         select
187             [Name], [CountryRegionCode]
188         from
189             [Sales].[SalesTerritory]
190         where
191             [Group]='North America'
192     </query>
193 </result-set>
194 </system-under-test>
195 <assert>
196     <row-count>
197         <equal>@NumeroSeis</equal>
198     </row-count>
199 </assert>
200 </test>
201 <test name="Usando una variable en la query obtenemos el número
de ventas por territorio (Error - Se espera 3)">
202     <system-under-test>
203         <result-set>
204             <query>
205                 select
206                     [Name], [CountryRegionCode]
207                 from
208                     [Sales].[SalesTerritory]
209                 where
210                     [Group]='Europe'
211             </query>
212         </result-set>
213     </system-under-test>
214     <assert>
215         <row-count>
216             <equal>@NumeroSeis</equal>
217         </row-count>
218     </assert>
219 </test>
220 <test name="El número de ventas por territorio es mayor que el
valor devuelto por la query (Correcto - Assert devuelve 1)">
221     <system-under-test>
222         <result-set>
223             <query>
224                 select
225                     [Name], [CountryRegionCode]
226                 from
227                     [Sales].[SalesTerritory]
228                 where

```



```

229         [Group]='Europe'
230     </query>
231 </result-set>
232 </system-under-test>
233 <assert>
234     <row-count>
235         <more-than>
236             <query-scalar>
237                 select
238                 count(*)
239                 from
240                 [Sales].[SalesTerritory]
241                 where
242                 [Group] = 'Europe'
243                 and [Name] like 'F%'
244             </query-scalar>
245         </more-than>
246     </row-count>
247 </assert>
248 </test>
249 <test name="Columna VacationHours de la tabla
HumanResources.Employee es varchar (Error - Tipo smallint)">
250     <system-under-test>
251         <data-type>
252             <column caption="VacationHours"
perspective="HumanResources" table="Employee"/>
253         </data-type>
254     </system-under-test>
255     <assert>
256         <is>
257             nvarchar
258         </is>
259     </assert>
260 </test>
261 <test name="Columna NationalIDNumber de la tabla
HumanResources.Employee es varchar(15) (Correcto)">
262     <system-under-test>
263         <data-type>
264             <column caption="NationalIDNumber"
perspective="HumanResources" table="Employee"/>
265         </data-type>
266     </system-under-test>
267     <assert>
268         <is>
269             nvarchar(15)
270     </is>

```

```

271     </assert>
272 </test>
273 <test name="Columna 'HireDate' de la tabla
HumanResources.Employee es de tipo Date (Correcto)">
274     <system-under-test>
275         <data-type>
276             <column caption="HireDate" perspective="HumanResources"
table="Employee"/>
277         </data-type>
278     </system-under-test>
279     <assert>
280         <is>date</is>
281     </assert>
282 </test>
283 <test name="Columna 'HireDate' de la tabla
HumanResources.Employee es de tipo Datetime (Error - Tipo date)">
284     <system-under-test>
285         <data-type>
286             <column caption="HireDate" perspective="HumanResources"
table="Employee"/>
287         </data-type>
288     </system-under-test>
289     <assert>
290         <is>datetime</is>
291     </assert>
292 </test>
293 <test name="Columna 'Weight' de la tabla Production.Product es de
tipo 'decimal(8, 2)' (Correcto)">
294     <system-under-test>
295         <data-type>
296             <column caption="Weight" perspective="Production"
table="Product"/>
297         </data-type>
298     </system-under-test>
299     <assert>
300         <is>decimal(8, 2)</is>
301     </assert>
302 </test>
303 <test name="Keys como Date (Correcto)">
304     <system-under-test>
305         <result-set>
306             <query>
307                 <![CDATA[
308                     select cast('2018-08-30' as date) as [key], 100.0 as
[value]
309                     union all select cast('2018-08-31' as date) as [key], 77 as

```

```

310     [value]
311     ]]>
312     </query>
313 </result-set>
314 </system-under-test>
315 <assert>
316   <equalTo>
317     <column index="0" role="key" type="dateTime"/>
318     <query>
319       <![CDATA[
320         select cast('30/08/2018' as varchar(10)) as [key],
321         100.0 as [value]
322         union all select cast('31/08/2018' as varchar(10)) as
323         [key], 77 as [value]
324       ]]>
325     </query>
326   </equalTo>
327 </assert>
328 </test>
329 <test name="Keys como Date (Error)">
330   <system-under-test>
331     <result-set>
332       <query>
333         <![CDATA[
334           select cast('2018-08-30' as date) as [key], 100.0
335           as [value]
336           union all select cast('2018-08-31' as date) as [key], 77 as
337           [value]
338         ]]>
339       </query>
340     </result-set>
341 </system-under-test>
342 <assert>
343   <equalTo>
344     <column index="0" role="key" type="dateTime"/>
345     <query>
346       <![CDATA[
347         select cast('30082018' as varchar(10)) as [key],
348         100.0 as [value]
349         union all select cast('31082018' as varchar(10)) as [key],
350         77 as [value]
351       ]]>
352     </query>
353   </equalTo>
354 </assert>
355 </test>

```

```

349 <test name="Date como DateTime (Error)">
350   <system-under-test>
351     <result-set>
352       <query>
353         <![CDATA[
354           select cast('2015-12-01' as date) as [key], 100.0
355           as [value]
356           union all select cast('2015-12-02' as date) as [key], 77 as
357           [value]
358         ]]>
359       </query>
360     </result-set>
361   </system-under-test>
362   <assert>
363     <equalTo>
364       <column index="0" role="key" type="dateTime"/>
365       <query>
366         <![CDATA[
367           select cast('12/1/2015 08:16:52' as datetime) as
368           [key], 100.0 as [value]
369           union all select cast('12/2/2015 08:45' as datetime) as
370           [key], 77 as [value]
371         ]]>
372       </query>
373     </equalTo>
374   </assert>
375 </test>
376 </testSuite>

```

Listing B.1: Test de prueba básico con NBI

Lista de acrónimos

- AICPA** Instituto Americano de Contadores Públicos. 46
- BB.DD.** Base de datos. 12, 13, 48, 51, 54, 57
- BI** Business Intelligence. 1–3, 11, 13–18, 20–27, 31, 32, 34–36, 39, 41–46, 62–64
- BIMM** Business Intelligence Maturity Model. 26
- BSC** Balanced Scorecard. 21
- CMI** Cuadro de Mando Integral. 21
- COBIT** Control Objectives for Information and related Technology. 26
- CRISP-DM** Cross-Industry Standard Process for Data Mining. 28
- CRM** Customer Relationship Management. 19, 27
- CSV** Comma Separated Values. 53, 58
- ERP** Enterprise Resource Planning. 19
- ETL** Extract, Transform and Load. 2, 19, 20, 31, 32, 34, 35, 40–45, 62, 64
- IFRS** International Financial Reporting Standard. 47
- ISACA** Information Systems Audit and Control Association. 26, 84
- IT** Informartion Technologies. 7, 24, 25, 31, 33, 67
- ITGI** IT Governance Institute. 26, 84
- KPI** Key Performance Indicator. 20, 32

- NoSQL** Not Only Structured Query Language. 35, 42, 45
- ODBC** Open Database Connectivity. 35, 45
- OLAP** On-line Analytical Processing. 20, 27, 32, 34, 43–45, 62, 66
- OLEDB** Object Linking and Embedding for Databases. 35, 45
- RR.HH.** Recursos Humanos. 19
- SCM** Supply Chain Management. 27
- SQL** Structured Query Language. 42, 43, 47, 50, 72
- SSAS** SQL Server Analysis Services. 45
- SSIS** Microsoft SQL Server Integration Services. 45
- SSRS** SQL Server Reporting Services. 45
- XML** Extensible Markup Language. 43, 54, 56, 67, 68

Glosario

COBIT: Control Objectives for Information and related Technology Es un conjunto de mejores prácticas para el manejo de la información que fue creado por [Information Systems Audit and Control Association \(ISACA\)](#) e [IT Governance Institute \(ITGI\)](#) en el año 1992. 26

CRISP-DM: Cross-Industry Standard Process for Data Mining Es una metodología estándar no propietaria que fue propuesta por un consorcio europeo a finales de los 90 para la implementación de proyectos de minería de datos. 28

Excel Caos Problema resultante del uso excesivo de Excel como herramienta de análisis. Al tener cada usuario un fichero personalizado, existe un coste muy alto al querer sincronizar todos los ficheros para que cuadren y se puedan utilizar en la toma de decisiones. 16

Gartner Es una consultora internacional especializada en Tecnologías de Información y Comunicación, www.gartner.com, enero 2006. 14, 26

Hyper Cycle Representación gráfica de la madurez, adopción y aplicación de negocio de una o varias tecnologías específicas. 26

IFRS: International Financial Reporting Standard Son una serie de normas que tienen como objetivo establecer una base común para la presentación e interpretación de información financiera a nivel mundial. 47

Bibliografía

- [1] L. M. Z. Pérez, H. M. Hernández, and R. C. O. Mass, “Inteligencia de los negocios. clave del Éxito en la era de la información,” 2016, consultado el, 28 julio 2023. [En línea]. Disponible en: https://crunia.udc.gal/permalink/34CISUG_UDC/1mbsdp7/cdi_dialnet_primary_oai_dialnet_unirioja_es_ART0001117059
- [2] K. C. Laudon and J. P. Laudon, *Sistemas de información gerencial organización y tecnología de la empresa conectada en red*, 8th ed. México : Pearson Educación, 2004, consultado el, 27 julio 2023.
- [3] J. C. Díaz, “Introducción al business intelligence,” 2014, consultado el, 13 marzo 2023. [En línea]. Disponible en: https://crunia.udc.gal/permalink/34CISUG_UDC/1q8do7i/alma991005787578607714
- [4] R. Ingle, “How to conduct effective testing of business intelligence applications,” consultado el, 28 julio 2023. [En línea]. Disponible en: <https://www.bitwiseglobal.com/en-us/how-to-conduct-effective-testing-of-business-intelligence-applications/>
- [5] Z. R. Jiménez, “Testing en business analytics,” 2020, consultado el, 29 julio 2023. [En línea]. Disponible en: <https://blogvisionarios.com/articulos-data/testing-en-business-analytics/>
- [6] J. Sutherland and K. Schwaber, “Scrum guide,” consultado el, 1 marzo 2023. [En línea]. Disponible en: <https://scrumguides.org/scrum-guide.html>
- [7] F. Heath, *The Professional Scrum Master (PSM I) Guide: Successfully practice Scrum with real-world projects and achieve your PSM I certification with confidence*. Packt Publishing, 2021, consultado el, 2 marzo 2023.
- [8] J. Martins, “Diagrama de gantt: qué es y cómo crear uno con ejemplos,” 2022, consultado el, 5 marzo 2023. [En línea]. Disponible en: <https://asana.com/es/resources/gantt-chart-basics>

- [9] “Boe-a-2023-6346 (resolución de 27 de febrero de 2023, de la dirección general de trabajo, por la que se registra y publica el xx convenio colectivo nacional de empresas de ingeniería; oficinas de estudios técnicos; inspección, supervisión y control técnico y de calidad),” 2022, consultado el, 4 agosto 2023. [En línea]. Disponible en: [https://www.boe.es/eli/es/res/2023/02/27/\(6\)](https://www.boe.es/eli/es/res/2023/02/27/(6))
- [10] J. L. Cano, *Business Intelligence: Competir con información*, 1st ed. Banesto, Fundación Cultur [i.e. Cultural], 2007, consultado el, 13 marzo 2023.
- [11] Y. Sierra, “Business intelligence: todo lo que necesitas saber y sus ejemplos,” 2020, consultado el, 13 marzo 2023. [En línea]. Disponible en: <https://blog.lemontech.com/business-intelligence-concepto-caracteristicas-software-y-ejemplos/>
- [12] C. Romero, “Componentes de un sistema de business intelligence,” 2016, consultado el, 25 marzo 2023. [En línea]. Disponible en: <https://labiblioteca.com/2016/11/07/componentes-de-un-sistema-de-business-intelligence/>
- [13] R. S. Kaplan and D. P. Norton, *El Cuadro de Mando Integral*, 3rd ed. Gestion 2000, Grupo Planeta, 2009, consultado el, 14 marzo 2023.
- [14] A. Camps, “8 beneficios del business intelligence para tu empresa,” 2023, consultado el, 13 marzo 2023. [En línea]. Disponible en: <https://www.bytemaster.es/nuevas-tecnologias/business-intelligence-8-beneficios-para-tu-empresa/>
- [15] IBM, “What is data mining?” consultado el, 10 abril 2023. [En línea]. Disponible en: <https://www.ibm.com/topics/data-mining>
- [16] L. J. Aguilar, *Sistemas De Información En La Empresa*, 1st ed. Alfaomega, 2015, consultado el, 3 abril 2023.
- [17] “Proceso del data mining,” consultado el, 31 julio 2023. [En línea]. Disponible en: <https://www.ceupe.com/blog/proceso-del-data-mining.html>
- [18] F. Toledo, “Testing en business intelligence,” 2017, consultado el, 13 abril 2023. [En línea]. Disponible en: <https://www.federico-toledo.com/testing-en-business-intelligence/>
- [19] R. KeepCoding, “5 tipos de pruebas o testing de software,” 2022, consultado el, 19 abril 2023. [En línea]. Disponible en: <https://keepcoding.io/blog/tipos-de-pruebas-o-testing-de-software/>
- [20] IEBSchool, “Tipos de licencias de software: Todo lo que tienes que saber,” 2023, consultado el, 18 abril 2023. [En línea]. Disponible en: <https://www.iebschool.com/blog/modelos-negocios-software-libre-open-source-digital-business/>

- [21] “Motiopi,” consultado el, 20 abril 2023. [En línea]. Disponible en: <https://motio.com/es/products/motiopi/>
- [22] C. L. Charlier, “Nbi is an open-source framework to test your business intelligence solutions or validate your data quality,” consultado el, 24 abril 2023. [En línea]. Disponible en: <http://www.NBi.io/>
- [23] “Las 10 mejores herramientas de prueba de sap (herramientas de automatización de sap),” consultado el, 16 mayo 2023. [En línea]. Disponible en: https://spa.myservername.com/top-10-best-sap-testing-tools#2_RightData
- [24] “What is querysurge?” consultado el, 18 mayo 2023. [En línea]. Disponible en: <https://www.querysurge.com/product-tour/what-is-querysurge>
- [25] “Bi validator – bi testing tool key features benefits,” consultado el, 19 mayo 2023. [En línea]. Disponible en: <https://www.datagaps.com/bi-testing-tools/bi-validator/>
- [26] A. Cartledge, “What is bi testing?” 2023, consultado el, 19 abril 2023. [En línea]. Disponible en: <https://wiiisdom.com/blog/what-is-bi-testing/>
- [27] M. Tadayon, “Tmf: a minimal testing tool for ruby,” consultado el, 15 mayo 2023. [En línea]. Disponible en: <https://github.com/browsersenior/tmf>
- [28] “Bi testing,” consultado el, 20 abril 2023. [En línea]. Disponible en: <https://www.datagaps.com/data-testing-concepts/bi-testing/>
- [29] K. E. Vimos-Buenaño and M. M. T. Palacios, “Contabilidad administrativa ¿qué es y cómo aplicarla en organizaciones de la economía popular y solidaria?” *Cienciamatria*, 2021, consultado el, 31 julio 2023. [En línea]. Disponible en: https://crunia.udc.gal/discovery/fulldisplay?docid=cdi_dialnet_primary_oai_dialnet_unirioja_es_ART0001514999&context=PC&vid=34CISUG_UDC:CRUNIA&lang=gl&search_scope=MyInst_and_CI&adaptor=Primo%20Central&tab=Everything&query=any,contains,que%20es%20la%20contabilidad&offset=0
- [30] J. A. A. Ensuncho, “Qué es y qué no es contabilidad,” *Lúmina*, 1998, consultado el, 31 julio 2023. [En línea]. Disponible en: https://crunia.udc.gal/discovery/fulldisplay?docid=cdi_dialnet_primary_oai_dialnet_unirioja_es_ART0001514999&context=PC&vid=34CISUG_UDC:CRUNIA&lang=gl&search_scope=MyInst_and_CI&adaptor=Primo%20Central&tab=Everything&query=any,contains,que%20es%20la%20contabilidad&offset=0

- [31] “Cierre contable y financiero,” consultado el, 31 julio 2023. [En línea]. Disponible en: <https://www.sap.com/spain/products/financial-management/accounting-financial-close.html>
- [32] “Sap fico: ¿para qué se utiliza?” consultado el, 31 julio 2023. [En línea]. Disponible en: <https://www.cursosapoficial.com/sap-fico-para-que-se-utiliza/>
- [33] L. Rodríguez, “10 buenas prácticas de testing,” consultado el, 29 julio 2023. [En línea]. Disponible en: <https://luixrodriguezneches.wordpress.com/2009/07/26/10-buenas-practicas-de-testing/>