ISSN: 2594 - 2905

IMPLEMENTATION OF A BUSINESS INTELLIGENCE MODEL FOR SMEs USING MICROSERVICES

IMPLEMENTACION DE UN MODELO DE INTELIGENCIA DE NEGOCIOS PARA PYMES USANDO MICROSERVICIOS

Romero Castro Javier [1], Arias Peregrino Víctor Manuel [2], León de la O Dulce María [3], Del Ángel Delgado Hugo [4], Valdez Ramos María de la Luz [5]

https://doi.org/10.61117/ipsumtec.v8i1.356

[1] Civil Engineer, Specialist in Informatics, National Technological Institute of Mexico/Technological Institute of Villahermosa, Division of Graduate Studies and Research, 9931604868, M23301207@villahermosa.tecnm.mx, ZIP Code 86010, https://orcid.org/0009-0004-7461-4803.
[2] Doctor in Technological Development, National Technological Institute of Mexico/Technological Institute of Villahermosa, Division of Graduate Studies and Research, 9365937388, victor.ap@villahermosa.tecnm.mx, ZIP Code 86010, https://orcid.org/0000-0001-5925-4097.
[3] Master's in Systems Engineering, National Technological Institute of Mexico/Technological Institute of Villahermosa, Division of Graduate Studies and Research, 9931750754, dulce.leono@villahermosa.tecnm.mx, ZIP Code 86010, https://orcid.org/0000-0003-0877-7861.
[4] Master's in Systems Engineering, National Technological Institute of Mexico/Technological Institute of Villahermosa, Division of Graduate Studies and Research, 9932287573, hugo.ad@villahermosa.tecnm.mx, ZIP Code 86010, https://orcid.org/0009-0002-1374-9478.
[5] Doctor in Technological Development, National Technological Institute of Mexico/Technological Institute of Villahermosa, Division of Graduate Studies and Research, 9931922043, maria.valdezr@villahermosa.tecnm.mx, ZIP Code 86010, https://orcid.org/0000-0002-8726-5874.

Abstract -- This article proposes the feasibility of developing a business intelligence model for SMEs through web technologies, a microservices architecture, and Python for data analysis. The structure is based on the use of web services following the REST (Representational State Transfer) architecture, allowing adaptable integration of applications and data through autonomous microservices. The proposed architecture is grounded in the use of web technologies, ensuring an intuitive and user-friendly experience and compatibility between systems. Python was selected for its adaptability and dedicated data science libraries, facilitating data analysis to help small and medium-sized enterprises transform data into strategic information to improve processes and gain a competitive advantage.

Keywords: *Data Analysis, Business Intelligence, Microservices, SMEs, REST.*

INTRODUCTION

Business intelligence and data analytics in business processes represent a competitive advantage factor, which requires using technological tools and defining processes that allow the creation of mature management models. Business intelligence and data analytics are essential as a business strategy, significantly impacting performance and decision-making (Baron Ramirez, García Estrella, & Sánchez Gárate, 2021).

The term Business Intelligence (BI) was defined by Gartner in the mid-1990s, although the concept originates in the development of Management Information Systems (MIS) in the 1970s, as companies began to automate tasks (Joyanes, 2019). Key tools for business intelligence include database query and reporting software, tools for multidimensional data analysis (online analytical processing), and data mining tools (Laudon & Laudon, 2012).

BI and data analysis have become essential instruments for companies of all sizes, enabling operational optimization, market trend detection, and customer satisfaction improvement. However, for small and medium-sized enterprises (SMEs), implementing these technologies remains challenging due to limited financial resources, a lack of specialized personnel, and the technical complexity of conventional BI systems.

Today, advanced models, technologies, and software allow companies to utilize BI solutions tailored to their needs and constraints, yet these models often focus on large corporations or are challenging to integrate with SME computing systems. As noted by Rosen, Lublinsky, Smith, & Balcer (2008), the integration of existing applications and data is one of the most significant challenges businesses face. Too often, fragile and unsustainable solutions are implemented, creating a complex web of connections using a wide variety of technologies and protocols.

This document aims to demonstrate the feasibility of using a simple methodology and software architecture to create a practical BI model for SMEs, enabling them to convert data into valuable information for strategic decision-making. The technologies used ensure smooth and scalable integration with established systems in companies that have developed their solutions. This

scope includes implementing an accessible architecture, using fundamental technologies for data analysis, and tools that foster interoperability between systems.

The implementation of the BI model for SMEs was carried out in a small company called Financiera Tecnológica de Chiapas (FINTECH), which provides short-term loans to low-income individuals. This company meets all the requirements for implementing the model (Romero et al., 2024).

Despite its extensive experience in the financial sector and a strong client portfolio, the company faces challenges in optimizing its credit evaluation and risk management processes. Developing a BI model offers an opportunity to advance the application of these new technologies and achieve its objectives more efficiently (Romero et al., 2024).

This work presents basic examples of the methodology and techniques used to generate a BI module and integrate it into the company's client and loan management system. It allows reliability evaluations for new clients and renewal assessments based on clients' credit history, as well as performance analyses for advisors, portfolios, and regions.

DEVELOPMENT

According to Ralph Kimball, the methodology is based on four fundamental principles (Forero-Castañeda & Sánchez-Garcia, 2021):

- Have an in-depth understanding of the business to implement the model based on a well-defined set of requirements.
- Build an adequate information infrastructure.
- Deliver in meaningful increments, ensuring they are neither too extensive nor too brief.
- Provide a complete solution, including query tools, graphical applications for reporting, training, and support, depending on the technology chosen for implementation.

For the implementation of a BI model for SMEs, a strategy based on the stages recommended by (Accenture High Performance Center, 2008) is considered:

Stage: Organization and strategy
- Identification of the company's needs
- Diagnosis of the company's capabilities
- Management's decision to adopt the model
- Analysis and preparation of the business intelligence model
- Planning for the transformation of IT services

Stage: Tactical implementations
- Detailed definition of strategic components
- Adjustment of existing processes and organization
- Infrastructure
- Definition of the catalog of processes and services
- Definition of business management processes for the business intelligence model

Stage: Platform design
- Development of the model's strategic components
- Implementation of structural services and processes
- Infrastructure organization
- Process monitoring
- Implementation of the operational catalog of processes and services

Stage: General integration
- Guidelines for creating and implementing services
- Service reuse framework
- Integration of the business intelligence model into the company's processes and decision-making
- Model evaluation

Participants
The key participants in this proposal include:

- Clients: providing detailed socioeconomic information for evaluation and analysis.
- Advisors and Coordinators: whose activities will be assessed based on performance metrics.
- Administrative Staff: able to consult the general and detailed evaluations resulting from the business intelligence model.
- The Company's IT Team: responsible for implementing and maintaining the system.

Tools used

To implement and evaluate the proposal, the following tools will be used:

Current system: The current system, developed in PHP with a Model-View-Controller (MVC) architecture and a MariaDB database, allows the registration of key information such as clients, loans, payments, portfolios, advisors, and other essential data.

Figure 1. Client registration in the current system

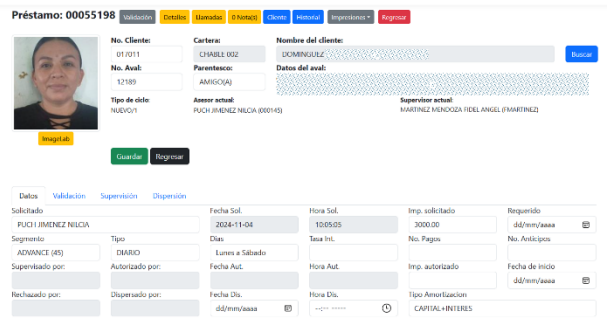

Figure 2. Loan registration in the current system

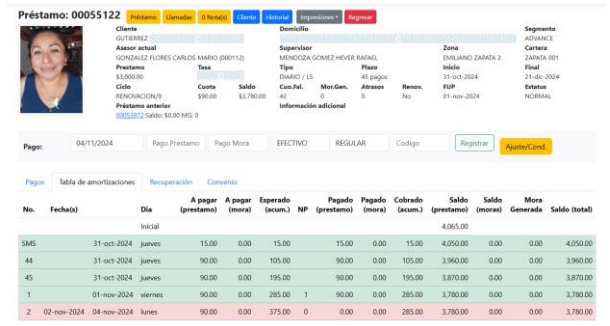

Figure 3. Payment registration in the current system

Prepared data warehouse for analysis (BDA): An independent database specifically designed to store and process information relevant to data analysis, thus preventing interference with the existing transactional database and, on the other hand, reducing the need for calculations with values stored in the database and maintaining relationships between tables. The BDA was created using MariaDB software, which is one of the most popular open-source relational databases. It was developed by the original creators of MySQL and is guaranteed to remain open-source software (MariaDB Foundation, 2019).

REST Microservices: Services developed in PHP and Python for transferring and processing data between the transactional database and the prepared data warehouse. PHP will be used for communication with the current system, while Python will be used for data analysis. Representational State Transfer (REST) is a software architecture created as a guide to manage communication in a complex network like the internet. It can be used to support reliable, high-performance communications at scale and can be easily implemented and modified, providing visibility and portability between platforms for any system (Amazon Inc., n.d.).

Together, the microservices form procedure interfaces in which existing programs or subsystems offer a variety of services, accessed by invoking the procedures of the interface. These interfaces are sometimes referred to as Application Programming Interfaces (APIs) (Sommerville, 2005).

Methodology and Procedure

Determination of metrics

According to the definition proposed by Joyanes (2019), a metric is a quantitative measure that allows one to assess the status of a website, a webpage, or a process conducted on a website, for a specific attribute or parameter. For the purposes of this work, a metric is defined as a quantifiable indicator used to assess the performance, properties, or results of a process.

For a financial company that grants loans, a metric can evaluate crucial aspects of its operations, such as assessing a new customer based on factors like economic activity, age, gender, education level, etc., loan approval rates, liquidation times, or the default rate of granted credits. The number and type of metrics required for a BI model are determined solely by the needs of the company, and there is no specific pattern or limit to this.

To determine a relevant metric for the company under study, the following steps were followed:

1. Describe the Objective of the Metric: Identify the purpose or need that the metric aims to fulfill and understand what valuable information is to be obtained through the metric.
2. Determine the Data Required to Calculate the Metric: Identify the data or variables needed for calculating the metric and ensure that these data are available or can be collected.
3. Determine the Calculation Method for the Metric: Define the formula or algorithm to be used to calculate the metric, ensuring it is correct and yields the desired results.
4. Describe the Interpretation of the Results of a Metric: Establish the meaning or relevance of the values returned by the metric, understand what a high or low value indicates, and how it relates to the objective.
5. Create a Result Valuation Table for the Metric: Develop a table associating the metric values with their interpretation, defining thresholds, ranges, or scales that allow categorization and interpretation of the results.
6. Determine the Weighting of the Metric: Assign a relative weight or importance to the metric within a set of metrics and establish how it will be integrated with other metrics to derive a KPI of interest to the company. KPIs (Key Performance Indicators) are metrics used to quantify objectives, reflecting performance and progress toward achieving those objectives (Joyanes, 2019).

Example of metric determination

Metric: Default Rate for People with "n" Dependents

Description: This metric calculates the default rate among clients who have "n" dependents, as having additional dependents can affect the ability to repay. This metric can be calculated by geographic area.

Required Data: Geographic area, number of dependents, clients in the area, loans in the area.

Formula:

$$T_{z,n} = \frac{P_{\text{unpaid},z,n}}{P_{\text{total},z}}$$

Where:

- z is the geographic zone to analyze.
- n is the number of economic dependents to analyze.
- $T_{z,n}$ is the default rate of loans in zone "z" for clients with "n" economic dependents.
- $P_{\text{unpaid},z,n}$ is the number of unpaid loans by clients with "n" economic dependents in zone "z".
- $P_{\text{total},z}$ is the total number of loans in zone "z".

Interpretation of results:
If the default rate among clients with "n" dependents is directly proportional to the level of risk, meaning that the higher the default rate, the higher the risk.

Table 1. Metric Valuation Table

Risk Level	Default Rate	Risk Factor	Interpretation
Low	0% - 5%	1	Very low risk; high compliance
Moderate	6% - 15%	2	Moderate risk; good history
Medium	16% - 25%	3	Medium risk; variable compliance
High	26% - 40%	4	High risk; frequent defaults
Very High	More than 40%	5	Very high risk; elevated default

Weighting: 10%

Table 2. Example of Metrics Integration (KPI Calculation)

Metric	Risk Factor	Weighting (%)	Weighted Score
Age	2	5%	0.1
Education	3	10%	0.3
Dependents	4	15%	0.6
Loan-to-Income Ratio	3	20%	0.6
Default Rate by Gender & Education	2	10%	0.2
Default Rate by Gender & Dependents	4	10%	0.4
Early Payoff Rate by Gender & Income	1	15%	0.15
Default History by ZIP & Income	3	15%	0.45
Key Performance Indicator		100%	2.8

Table 3. Interpretation of the Key Performance Indicator

KPI	Risk Level	Description
1.0 - 1.5	Very Low	Customer with very strong characteristics; minimal risk of default.
1.6 - 2.5	Low	Low risk; customer with good repayment capacity and history in similar areas.
2.6 - 3.5	Below Average	Below average risk; customer with some risk areas but generally stable.
3.6 - 4.5	Above Average	Above average risk; the customer presents various factors that increase the risk.
4.6 - 5.0	High	High risk; customer with multiple risk factors; high risk of default.

The value obtained from the key performance indicator was 2.8 for this example, meaning the new client is classified in a low-medium risk category (since the score is between 2.6 and 3.5 on a scale of 1 to 5). This means that although the client shows some risk characteristics, they are not high enough to consider them high risk. This type of client could qualify for the loan, but with adjusted conditions or additional monitoring.

General architecture of the proposed model

For the implementation of the current proposal, the architecture proposed by (Romero, et al., 2024) was used, which is as follows:

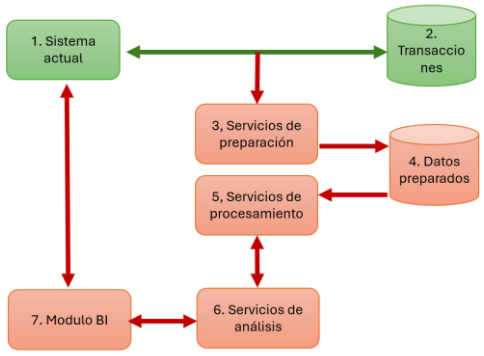

Figure 4. Simplified Architecture Proposal to Implement the Business Intelligence Model

This article focuses on showing examples of the implementation of components 3 to 7 and their integration into the existing system within the company.

Data warehouse prepared for analysis (BDA)

An independent database was created, isolated from the transactional data warehouse of the current system, which exclusively stores data relevant to the BI model. The

ISSN: 2594 - 2905

isolation of this warehouse ensures that it does not interfere with the operation of the current system, even in the event of failures or interruptions in the operation of the BI model.

For this data warehouse, an optimized schema was established to support data analysis, avoiding the inclusion of unnecessary information. In this exercise, the tables clients and loans were selected. For each of them, the fields that are likely to be used for the analysis and evaluation of the proposed BI model were chosen, as shown below:

Name	Type	Length	Comment
IdPrestamo	int	11	No. de prestamo
IdCliente	int	11	No. de cliente
IdAval	int	11	No. de aval del prestamo
IdZona	int	11	No. de zona
IdCartera	int	11	No. de cartera
IdAsesor	int	11	No. de asesor que registra la solicitud
IdCoordinador	int	11	No. de coordinador de zona
ImporteSolicitado	decimal	12	Importe solicitado
ImporteAutorizado	decimal	12	Importe autorizado
FechaInicial	date		Fecha de inicio
FechaFinal	date		Fecha de termino
FechaUltimoPago	date		Fecha de ultimo pago
NoPagos	int	11	No. de pagos
SaldoCapital	decimal	12	Saldo de capital
SaldoIntereses	decimal	12	Saldo de intereses
SaldoTotal	decimal	12	Saldo total
MorasGeneradas	int	11	No. de atrasos generados
Liquidado	tinyint	1	Liquidado (0=np,1=si)

Figure 5. Loans Table - BDA

Name	Type	Length	Comment
IdCliente	int	11	No. Cliente
IdCartera	int	11	No. Cartera
IdZona	int	11	No. Zona
IdAsesor	int	11	No. Asesor que registra al cliente
FechaNacimiento	date		Fecha de nacimiento
IdGenero	char	1	Genero
IdEstadoCivil	tinyint	1	Estado civil
IdEscolaridad	int	11	Escolaridad
IdActividad	int	11	Actividad
IdTipoActividad	int	11	Tipo de actividad
IdTipoPropiedadDom	int	11	Tipo de propiedad del domicilio
IdTipoPropiedadLoc	int	11	Tipo de propiedad del local
NoEmpleados	int	11	No. de empleados
RentaDomicilio	decimal	12	Importe de renta del domicilio
RentaLocal	decimal	12	Importe de renta del local
IdNivelIngresos	int	11	Nivel de ingresos
IdNivelEgresos	int	11	Nivel de egresos
IdNivelVentas	int	11	Nivel de ventas
DomTiempo	int	11	Tiempo de vivir en el domicilio
NegTiempo	int	11	Tiempo de dedicarse a su negocio
LocTiempo	int	11	Tiempo de usar el local actual
IdTipoUbicacion	int	11	Tipo de ubicacion del negocio
CodigoPostal	char	5	Codigo postal
NoDependientes	tinyint	4	No. de dependientes

Figure 6. Clients Table - BDA

Preparation services

These are REST microservices that implement the ETL layer of the proposed model. This layer focuses on three main processes: extraction, transformation, and loading of data. Extraction is the process of identifying and collecting relevant or significant data from various sources. Transformation is the process of converting the data using a set of business rules, and finally, the last phase of the process is loading the data into the target repository (Joyanes, 2019).

The services in this layer receive the information of a record (client or loan) from the current system, select the values of interest for the BI model, perform the necessary calculations, and register or update the selected values in the BDA.

For the case presented in this document, 2 preparation microservices were implemented in PHP, one for clients and the other for loans. These data include the socio-economic information of the clients, their payment history, and the information regarding zones, advisors, and portfolios.

```php
<?php
/*
    Microservicio: BI-registrar-cliente
    Descripción  : Guarda o actualiza un cliente
    Recibe       : Archivo JSON con los datos del cliente del sistema actual
    Devuelve     : Archivo JSON con el resultado de la operación
*/
// Incluir la conexion a la BDP
require_once('connection.php');

// Recibir los datos del POST
$data = $_POST;

// Campos de la tabla `clientes` del modelo BI
$clienteBiFields = array(
    "IdCartera", "IdZona", "IdAsesor", "FechaNacimiento", "IdGenero", "IdEstadoCivil", "IdEscolaridad",
    "IdActividad","IdTipoActividad", "IdTipoPropiedadDom", "IdTipoPropiedadLoc", "NoEmpleados",
    "RentaDomicilio", "RentaLocal", "IdNivelIngresos", "IdNivelEgresos", "IdNivelVentas", "DomTiempo",
    "NegTiempo", "LocTiempo", "IdTipoUbicacion", "CodigoPostal", "NoDependientes"
);

// Determinar si es un cliente nuevo o existente
$idCliente = $data['IdCliente'];

if ($idCliente == 0) {
    $sql = "INSERT INTO clientes (" . implode(", ", $clienteBiFields) . ")
    VALUES (?" . str_repeat(", ?", count($clienteBiFields) - 1) . ")";
    $stmt = $conn->prepare($sql);
    $params = array_map(function($field) use ($data) {
        return $data[$field];
    }, $clienteBiFields);
    $types = str_repeat("s", count($clienteBiFields));
} else {
    $sql = "UPDATE clientes SET " . implode(" = ?, ", $clienteBiFields) . " = ? WHERE IdCliente = ?";
    $stmt = $conn->prepare($sql);
    $params = array_merge(array_map(function($field) use ($data) {
        return $data[$field];
    }, $clienteBiFields), [$idCliente]);
    $types = str_repeat("s", count($clienteBiFields)) . "i";
}

call_user_func_array(array($stmt, "bind_param"), array_merge([$types], $params));

// Ejecutar la consulta
if ($stmt->execute()) {
    $response = array(
        "success" => true,
        "message" => "Operación exitosa"
    );
} else {
    $response = array(
        "success" => false,
        "message" => "Error: " . $stmt->error
    );
}

// Convertir la respuesta a JSON y enviarla
echo json_encode($response);

// Cerrar la conexión
$stmt->close();
$conn->close();
?>
```

Figure 7. Example of Preparation Microservice for Registering or Updating Clients in the BDA

Both services will be consumed from the current system via a call using the Curl library. For each of them, a simple function is created within the current system that receives the data in an array and sends it directly to the corresponding microservice, returning the response

obtained from the microservice. For example, to consume the microservice for registering or updating a client's data, the function is as follows:

```php
/*
    Funcion     : bi_registrar_cliente
    Descripción : Realizar  llamada al microservicio para registrar un cliente en la BDP
    Recibe      : Arreglo con los datos del cliente (POST)
    Devuelve    : Respuesta del microservicio
*/

Codeium: Refactor | Explain | Generate Function Comment | ×
function bi_registrar_cliente($data) {
    // URL del servicio web REST
    $url = "http://otro-servidor.com/api-php/bi-registrar-cliente.php";

    // Codificar los datos en formato JSON
    $jsonData = json_encode($data);

    // Configuración de la solicitud cURL
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_HTTPHEADER, [
        "Content-Type: application/json", // Indicar que el contenido es JSON
        "Content-Length: " . strlen($jsonData)
    ]);
    curl_setopt($ch, CURLOPT_POSTFIELDS, $jsonData);

    // Ejecutar la solicitud y obtener la respuesta
    $response = curl_exec($ch);

    // Cerrar cURL
    curl_close($ch);

    return($response);
}
```

Figure 8. Example of a function in the current system to consume a BI model preparation microservice

This function will be used each time a new client is registered, or the data of an existing client is updated, for example, in the controller corresponding to the client data:

```php
// guardar los datos recibidos del formulario en el objeto del modelo
$cliente->set_values($_POST);

if ($op== 'nuevo') {

    $newid=$cliente->insert();

    // BI: Registrar cliente
    $response = bi_registrar_cliente($_POST);

    // Ejemplo de codigo para validar la respuesta del microservicio
    if ($response == "false") {
        alert("No se pudo registrar el cliente en el modelo BI");
    }

    redirect($location_list);

} elseif ($op== 'actualizar') {

    $cliente->update($id);

    // BI: Registrar cliente
    $response = bi_registrar_cliente($_POST);

    // Ejemplo de codigo para validar la respuesta del microservicio
    if ($response == "false") {
        alert("No se pudo actualizar el cliente en el modelo BI");
    }

    redirect($location_list);
}
```

Figure 9. Customer registration code in the current system (modified to integrate the BI model)

Processing services

These are REST microservices in PHP designed to extract data from the BDA to the BI model. The number and functionality of these microservices are defined by the needs of the BI model and the metrics that are determined. A module of the BI model may make use of several microservices depending on what is required.

Just like the metrics, any number of services can be developed to retrieve all types of information from the BDA, depending on the needs of the BI model.

```php
<?php
/*
    Microservicio: BI-Clientes-Zona
    Descripción : Obtener todos los clientes de una zona
    Recibe      : Id de la zona
    Devuelve    : Archivo JSON con los datos de los clientes
*/
// Incluir la conexion a la BDP
require_once('connection.php');

$idzon = $_GET["id"];

// Consultar los registros de la base de datos
$sql = "SELECT * FROM clientes WHERE IdZona=" . $idzon;
$result = $conn->query($sql);

// Obtener los registros en un arreglo
$records = array();
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $records[] = $row;
    }
}

// Devolver los registros en formato JSON
//header('Content-Type: application/json');
echo json_encode($records);

// Cerrar la conexion a la base de datos
$conn->close();
?>
```

Figure 10. Example of a microservice for querying clients by zone

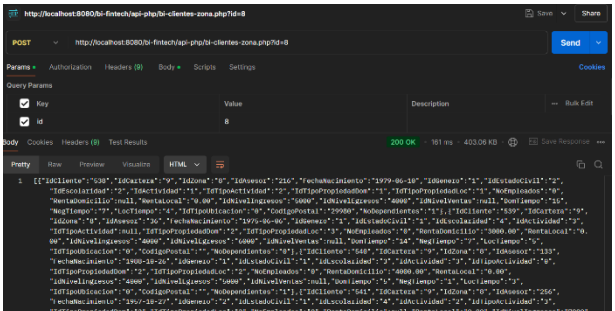

Figure 11. Example of results obtained by the BI-clientes-zona microservice

The results obtained through the processing microservices can be stored on the BI model server and configured to update only with the frequency deemed appropriate, with the aim of making the execution of data analysis more efficient.

Analysis Services

Once the metrics of interest for the company are determined, data analysis is carried out. To this end, REST microservices in Python were developed to perform queries, aggregation, and transformation of the data stored in the BDA through the formulas established for each metric.

Python is used for data analysis due to its ability to execute statistical models and algorithms that allow for data analysis (Python, n.d.).

Python has several features that make it ideal for learning (and doing) data science (Grus, 2023):

- It is free.
- It is relatively simple to write code (and particularly, to understand).
- It has many libraries associated with data science that are highly useful.

Additionally, the Pandas library is used, which is an open-source tool for data analysis and manipulation, fast, powerful, flexible, and easy to use, developed over the Python programming language (NumFOCUS Foundation, n.d.).

For publishing the REST services in Python, the FastAPI library is used, which is a modern and fast (high-performance) web framework for building APIs with Python, based on Python's standard type annotations (Sebastián Ramírez (@tiangolo), n.d.).

The microservices developed in this language will query and analyze data from the BDA through the processing services and return results to the existing system in real time or on demand.

Below is an example of a REST service that consumes a processing service, which provides the necessary data for analysis and returns the results of the integration of the defined metrics:

```python
import pandas as pd
import requests
from typing import Dict,List

Codeium: Refactor | Explain | Generate Docstring | ×
def obtener_clientes_zona(id_zona):
    url = f"http://localhost:8080/bi-fintech/api-php/bi-clientes-zona.php?id={id_zona}"
    try:
        response = requests.get(url)
        response.raise_for_status()  # Verifica si hubo un error en la solicitud
        datos = response.json()  # Convierte la respuesta a JSON
        return datos
    except requests.exceptions.RequestException as e:
        print(f"Error al obtener clientes de la zona {id_zona}: {e}")
        return None

Codeium: Refactor | Explain | Generate Docstring | ×
def obtener_prestamos_zona(id_zona):
    url = f"http://localhost:8080/bi-fintech/api-php/bi-clientes-zona.php?id={id_zona}"
    try:
        response = requests.get(url)
        response.raise_for_status()
        datos = response.json()
        return datos
    except requests.exceptions.RequestException as e:
        print(f"Error al obtener préstamos de la zona {id_zona}: {e}")
        return None
```

Figure 12. Example of REST service code in Python to evaluate a metric (1/2)

```python
Codeium: Refactor | Explain | Generate Docstring | ×
def calcular_tasa_incumplimiento(id_zona: str, no_dependientes: str, clientes_zona_df: pd.DataFrame,
    prestamos_cartera_df: pd.DataFrame) -> Dict[str, float]:

    clientes_filtrados = clientes_zona_df[(clientes_zona_df['IdZona'] == id_zona) &
                                          (clientes_zona_df['NoDependientes'] == no_dependientes)]

    if clientes_filtrados.empty:
        return {'Tasa de Incumplimiento': 0.0, 'Clientes Encontrados': [], 'Prestamos Sin Liquidar': 0}

    clientes = clientes_filtrados['IdCliente'].tolist()

    prestamos_filtrados = prestamos_cartera_df[prestamos_cartera_df['IdCliente'].isin(clientes)]

    p_total = prestamos_filtrados.shape[0]
    p_incumplidos = prestamos_filtrados[prestamos_filtrados['Liquidado'] == '0'].shape[0]

    if p_total == 0:
        return {'Tasa de Incumplimiento': 0.0, 'Clientes Encontrados': clientes, 'Prestamos Sin Liquidar': p_incumplidos}

    tasa_incumplimiento  = p_incumplidos / p_total

    nivel_riesgo = calcular_nivel_riesgo(tasa_incumplimiento)

    return {
        'Métrica': 'Tasa de incumplimiento para personas con n dependientes económicos ',
        'Tasa de Incumplimiento': tasa_incumplimiento ,
        **nivel_riesgo,
    }
```

Figura 1. Ejemplo de código de servicio REST en Python para evaluar una metrica (1/2)

This web microservice consumes data from an operation microservice called bi-clientes-zona and performs all necessary calculations to obtain the expected result. Once the expected result (default rate) is obtained, the results are prepared to be returned to the requesting process:

```python
Codeium: Refactor | Explain | Generate Docstring | ×
def calcular_nivel_riesgo(tasa_incumplimiento: float) -> Dict[str, str]:

    niveles_riesgo: List[Dict[str, any]] = [
        {"max_tasa": 0.05, "Nivel de Riesgo": "Bajo (1)", "Factor de Riesgo": "1",
        "Interpretación": "Muy bajo riesgo; alto cumplimiento"},
        {"max_tasa": 0.15, "Nivel de Riesgo": "Moderado (2)", "Factor de Riesgo": "2",
        "Interpretación": "Moderado riesgo; buen historial"},
        {"max_tasa": 0.25, "Nivel de Riesgo": "Medio (3)", "Factor de Riesgo": "3",
        "Interpretación": "Riesgo medio; cumplimiento variable"},
        {"max_tasa": 0.40, "Nivel de Riesgo": "Alto (4)", "Factor de Riesgo": "4",
        "Interpretación": "Riesgo alto; frecuentes incumplimientos"},
        {"max_tasa": float("inf"), "Nivel de Riesgo": "Muy Alto (5)", "Factor de Riesgo": "5",
        "Interpretación": "Muy alto riesgo; incumplimiento elevado"}
    ]

    for nivel in niveles_riesgo:
        if tasa_incumplimiento <= nivel["max_tasa"]:
            return {
                "Nivel de Riesgo": nivel["Nivel de Riesgo"],
                "Factor de Riesgo": nivel["Factor de Riesgo"],
                "Interpretación": nivel["Interpretación"]
            }

    return {}
```

Figure 13. Example of REST service code in Python to evaluate a metric (2/2)

In the following image, you can see the results obtained by the microservice:

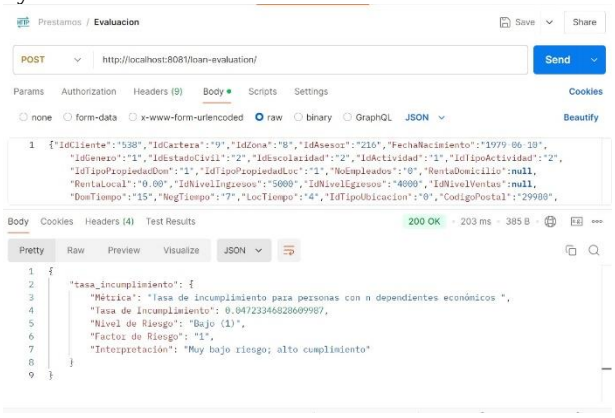

Figure 14. Example of metric calculation results

This process is repeated for each metric required and integrated to obtain a combined result of all the metrics for a performance indicator:

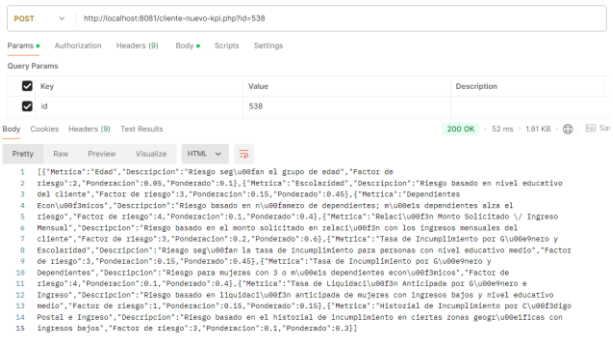

Figure 15. Example of integration of all metric results for a performance indicator

Model testing

The tests conducted on all the services of the Business Intelligence (BI) model were based on historical data from the company, with the aim of calibrating and assessing the accuracy of the metrics and the model itself. The company's staff plays a crucial role in this process, verifying the model's findings, as their experience is essential to ensure that the evaluations are relevant and aligned with the company's circumstances.

On the other hand, constant monitoring of the microservices is carried out to check their performance in real-time, facilitating efficient maintenance.

Furthermore, integration tests between the microservices and the existing system must be conducted to ensure proper data transmission and analysis, ensuring smooth and effective interoperability. One of the most important objectives of this work is to ensure minimal interference with the operation of the current system, which is one of the factors that often relegates the implementation of such technologies in proprietary systems.

Expected results

The results of the analyses, such as the reliability rating of a client or the evaluation of a zone, will be accessible from the current FINTECH system through an additional interface developed in PHP. Users will be able to query the BI model's results to enhance decision-making without leaving the main system.

The design includes specific screens that will display real-time BI analyses and performance metrics for advisors, portfolios, and zones. These interfaces will allow FINTECH to visualize key data in an accessible and practical manner for operational and strategic decisions.

The visualization interfaces for the BI model's results can be designed according to the company's needs and the current system's design. They can be displayed collectively or individually, summarized or detailed, in text or image format.

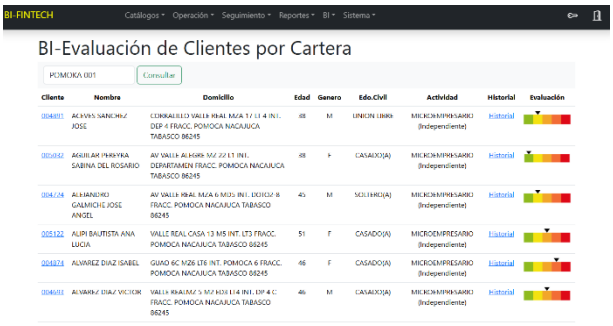

Figure 16. Example of client evaluation visualization

Figure 17. Example of detailed client evaluation

The BI model is open to developing any number of metrics and KPIs, applying various statistical and mathematical processes to make sense of the data recorded by the company. These results can be presented in various ways for interpretation by the users.

Below is an example that can be obtained using the Matplotlib library, which is a library for creating static, animated, and interactive visualizations in Python (Matplot Developing Team, n.d.):

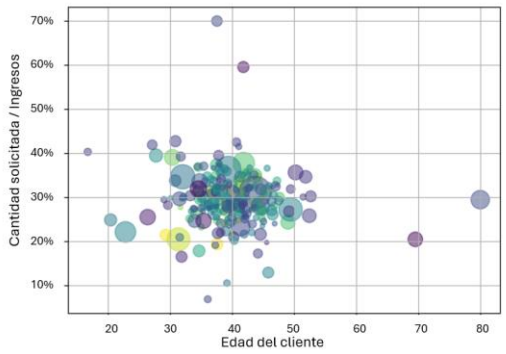

Figure 18. Example of visualization of results in a graphical format

The BI model for the Financial Technology SME of Chiapas is expected to achieve the following results:

- Loan Reliability: Evaluations of new clients with customized risk ratings and recommendations for loan distribution.

- Consultant and Area Evaluation: Performance indicators providing an accurate view of the performance of advisors and zone coordinators, assisting the company in improving its credit operations.
- Operational Effect: Agile and precise analysis processes, through the separation of the analysis database and the use of microservices, ensuring system scalability without compromising the current system's database.

DISCUSSION AND ANALYSIS OF RESULTS

In their study, (Gómez García, García Ruiz, & Acosta Gonzaga, 2020) explored the practical application of business intelligence (BI) in an SME with limited computer experience. By utilizing a free platform, they evidenced how BI facilitates the generation of crucial information for decision-making and deepens business understanding. Both this study and previous research conclude that BI represents a valuable resource for SMEs, whether through the development of customized technical solutions or the adoption of accessible platforms, allowing the conversion of data into knowledge to drive competitiveness.

The research by (Navarrete Reina, Samaniego Mena, & Oviedo Bayas, 2024) reveals key BI models and underscores the increasing importance of technology in information management. They also demonstrate a high regard for BI strategies due to their influence on economic growth and decision-making in SMEs. Consistently, both articles affirm that BI can be a potent resource for SMEs, attainable through either the development and seamless integration of systems into existing infrastructure or the implementation of BI strategies that foster adaptability, opportunity recognition, and robust financial standing.

In contrast, (Gutierrez Garcia, Figueroa Millan, Ochoa Ornelas, Chavez Valdez, & Arceo Dıaz, 2021) detail the development of a BI web module leveraging association rules, designed to augment a pre-existing decision support system (DSS). This approach utilizes data mining to provide insights into product supply and demand for a specific company. Although both research efforts aim to empower SMEs by converting data into actionable knowledge, their methodologies and case studies diverge. Specifically, one study highlights financial system integration through microservices, whereas the other centers on DSS augmentation through association rules and a web-based module.

The integration between the current system and the applied business intelligence (BI) model at the company “Financiera Tecnológica de Chiapas (FINTECH)” offers an example of the advantages that applying data science using exclusively open-source technologies can provide to small and medium-sized enterprises (SMEs).

When analyzing the performance of the proposed model, it was confirmed that the ability to produce objective evaluations of customer reliability and the performance of the company's portfolios and staff was effective. Testing verified that the proposed module is capable of handling large amounts of information without affecting the functionality of the current transaction system.

To adapt this technology to other platforms, careful consideration must be given to the technological infrastructure, the type of data being managed, and the specific purposes of each SME. This modular and adaptable method is particularly suitable for SMEs as it reduces costs and technical complexity, two major obstacles to implementing cutting-edge technologies in this sector.

CONCLUSIONS

This article presents a practical case of how a proprietary business intelligence model, based on microservices and data analysis with Python, can be developed to meet the demands of small and medium-sized enterprises. The results indicate that:

- The REST architecture of microservices facilitates scalability and enables effective integration with existing systems.
- Small and medium-sized enterprises have the capacity to transform operational data into strategic information, optimizing decision-making without requiring significant technological investments.
- The adaptability and flexibility of the model are crucial for its success across various contexts and company types.
- For the future of this proposal, advanced metrics can be developed and techniques such as regression models, predictive analysis, confidence intervals, correlation analysis, segmentation, clustering, etc., can be applied to allow SMEs to anticipate market changes and adjust their strategies in real-time.

Future research will focus on applying regression models, predictive analysis, and segmentation techniques to enhance analytical complexity, leveraging the model's capacity for statistical and mathematical data processing.

Further studies could investigate the long-term impact of implementing such business intelligence models in SMEs, evaluating their effects on strategic decision-making, operational efficiency, and overall competitiveness.

REFERENCES

[1] Baron Ramirez, E., García Estrella, C. W., & Sánchez Gárate, S. K. (2021). Business intelligence and data analytics. *Revista Científica de Sistemas e Informática*.
[2] Joyanes, L. (2019). *Business Intelligence and Data Analytics* (1st ed.). Marcombo.
[3] Laudon, K. C., & Laudon, J. P. (2012). *Management Information Systems*. Mexico City: Pearson.
[4] Rosen, M., Lublinsky, B., Smith, K., & Balcer, M. (2008). *Service-Oriented Architecture and Design Strategies*. Indianapolis, In.: Wiley Publishing, Inc.
[5] Romero, J., Arias, V. M., Leon, D. M., Del Angel, H., Valdes, M. d., & Hernandez, C. (2024). *Business Intelligence Model for SMEs: A Development Proposal*. *Innovación y Desarrollo Tecnológico Revista Digital*.
[6] Forero-Castañeda, D. A., & Sánchez-Garcia, J. A. (2021). *Introduction to Business Intelligence based on the Kimball Methodology*. *Tecnología, Investigación y Academia*.
[7] Accenture High-Performance Center. (2008). *Scala Learning*. Retrieved from Service-Oriented Architecture (SOA) How to Reformulate Corporate Architecture: https://gc.scalahed.com/recursos/files/r161r/w23913w/AccentureSOA.pdf
[8] MariaDB Foundation. (November 13, 2019). *Official MariaDB Page*. Retrieved January 16, 2024, from mariadb.org
[9] Amazon Inc. (n.d.). *What is a RESTful API?* Retrieved May 3, 2024, from https://aws.amazon.com/es/what-is/restful-api/
[10] Sommerville, I. (2005). *Software Engineering*. Madrid: Pearson.
[11] Python. (n.d.). *Official Python Page*. Retrieved February 8, 2024, from www.python.org
[12] Grus, J. (2023). *Data Science from Scratch, Basic Principles with Python*. Spain: O'Reilly - Anaya.
[13] NumFOCUS Foundation. (n.d.). *Official Pandas Page*. Retrieved September 9, 2024, from https://pandas.pydata.org/
[14] Sebastián Ramírez (@tiangolo). (n.d.). *FastAPI framework*. Retrieved September 18, 2024, from https://fastapi.tiangolo.com/
[15] Matplot Developing Team. (n.d.). *Matplotlib: Visualization with Python*. Retrieved October 18, 2024, from https://matplotlib.org/
[16] Gómez García, R., García Ruiz, A., & Acosta Gonzaga, E. (2020). *Benefits for SMEs through Business Intelligence: Case Study in a Social Event Services SME*. Revista Multidisciplinaria de Avances de Investigación.
[17] Navarrete Reina , V., Samaniego Mena, E., & Oviedo Bayas, B. (2024). *Strategies of Business Intelligence Models in the economic growth of SMEs*. Ciencia Huasteca Boletín Científico De La Escuela Superior De Huejutla, 12(24), 1-12. https://doi.org/10.29057/esh.v12i24.12503
[18] Gutierrez Garcia, L. T., Figueroa Millán, P. E., Ochoa Ornelas, R., Chávez Valdez, R. E., & Arceo Díaz, S. (2021). *Modeling of a Business Intelligence Web module based on association rules for the marketing of ornamental plants*. Difu100ci@, Revista de difusión científica, ingeniería y tecnologías, 15(3), 147-154. Retrieved October 18, 2024 from http://difu100cia.uaz.edu.mx/index.php/difuciencia/article/view/214

Contribution role	Definition
Conceptualization	Ma. de la Luz Valdez Ramos
Methodology	Dulce María León de la O
Project Management	Victor Manuel Arias Peregrino
Software	Javier Romero Castro
Supervision	Victor Manuel Arias Peregrino
Validation	Hugo del Angel Delgado
Visualization	Dulce María León de la O
Writing	Javier Romero Castro