



TECNOLÓGICO
NACIONAL DE MÉXICO®



Instituto Tecnológico de Culiacán

Tema Selecto de Base de Datos

Alumnos:

Abitia Burgos Julio Alejandro

Ramos Matunaga Raúl Alejandro

Torres Lizárraga Yair

Carrera:

Ingeniería en Sistemas Computacionales

Docente:

Dr. García Gerardo Clemente

Limpieza de Datos

Se definen las structs Record y Transaction para modelar la estructura de datos de las filas de los archivos solicitudes.csv y transacciones.csv, respectivamente.

```
struct Record {
    red: String,
    client_name: String,
    country: String,
    application_date: NaiveDate,
    employee_name: String,
    placed: String,
    card_id: u32,
    target: u32,
    gender: String,
}

struct Transaction {
    card_id: u32,
    date: NaiveDate,
    country: String,
    import: f64,
    usd: f64,
}
```

Tanto Record como Transaction implementan un método asociado from_str. Esta función actúa como un constructor que parsea una línea de texto completa del CSV y la convierte en una instancia de la struct correspondiente, asignando cada parte al campo apropiado.

```
impl Record {
    fn from_str(s: &str) -> Self {
        let parts = s.split(',').collect::<Vec<_>>();
        Self {
            red: parts[0].to_string(),
            client_name: parts[1].to_string(),
            country: parts[2].to_string(),
            application_date: normalize_date(parts[3]),
            employee_name: parts[4].to_string(),
            placed: parts[5].to_string(),
            card_id: parts[6].parse().unwrap(),
            target: parts[7].parse().unwrap(),
            gender: parts[8].to_string(),
        }
    }
}

impl Transaction {
    fn from_str(s: &str) -> Self {
        let parts = s.split(',').collect::<Vec<_>>();
        Self {
            card_id: parts[0].parse().unwrap(),
            date: normalize_date(parts[1]),
            country: parts[2].to_string(),
            import: parts[3].parse().unwrap(),
            usd: 0f64,
        }
    }
}
```

```

        }
    }
}

```

Para la operación inversa, ambas structs implementan el trait `Display`. Esto permite formatear una instancia de la struct directamente como una cadena de texto, con sus campos separados por comas, lo cual facilita su escritura directa en los archivos CSV de salida.

```

impl std::fmt::Display for Record {
    fn fmt(&self, f: &mut std::fmt::Formatter<'_>) -> std::fmt::Result {
        write!(f,
               "{} , {} , {} , {} , {} , {} , {} , {} , {} , {}",
               self.red,
               self.client_name,
               self.country,
               self.application_date.format("%-d/%-m/%-Y"),
               self.employee_name,
               self.placed,
               self.card_id,
               self.target,
               self.gender
        )
    }
}

impl std::fmt::Display for Transaction {
    fn fmt(&self, f: &mut std::fmt::Formatter<'_>) -> std::fmt::Result {
        write!(f,
               "{} , {} , {} , {} , {}",
               self.card_id,
               self.date.format("%-d/%-m/%-Y"),
               self.country,
               self.import,
               self.usd
        )
    }
}

```

Se implementa una función de utilidad, `normalize_date`, responsable de la normalización de las cadenas de fecha al tipo `NaiveDate`. Esta función maneja fechas anómalas mediante un bucle iterativo que decremente el día hasta encontrar una fecha válida, asegurando la integridad de los datos temporales.

```

fn normalize_date(date: &str) -> NaiveDate {
    let parts = date.split('/').collect::<Vec<_>>();

```

```

let mut day = parts[0].parse::<u32>().unwrap();
let month = parts[1].parse::<u32>().unwrap();
let year = parts[2].parse::<i32>().unwrap();

loop {
    match NaiveDate::from_ymd_opt(year, month, day) {
        Some(date) => return date,
        None => day -= 1,
    }
}
}

```

La función `main` orquesta todo el proceso en dos partes:

Primero lee el archivo de solicitudes. Normaliza los datos de los empleados para que todos los registros asociados a un mismo empleado en una misma fecha usen el objetivo más alto reportado para esa fecha. Luego, escribe los datos limpios en un nuevo archivo de salida.

```

let file = std::fs::File::open("solicitudes.csv").unwrap();
let file = std::io::BufReader::new(file);
let mut file = file.lines();

let mut seen: HashMap<u128, (String, HashMap<NaiveDate, u32>>) = HashMap::new();

let headers = file.next().map(|r| r.unwrap()).unwrap();

let data = file
    .map(Result::unwrap)
    .map(|row| Record::from_str(&row))
    .map(|row| {
        let hash: u128 = row
            .employee_name
            .split(' ')
            .map(|s| {
                let mut hasher = DefaultHasher::new();
                s.hash(&mut hasher);
                hasher.finish() as u128
            })
            .sum();
        seen.entry(hash)
            .and_modify(|(name, targets)| {
                targets
                    .entry(row.application_date)
                    .and_modify(|target| {
                        if *target < row.target {
                            *name = row.employee_name.clone();
                            *target = row.target
                        }
                    })
            })
    })

```

```

        })
        .or_insert(row.target);
    })
    .or_insert_with(|| {
        (
            row.employee_name.clone(),
            HashMap::from([(row.application_date, row.target)]),
        )
    });
    (hash, row)
})
.collect::<LinkedList<_, _>>();
let data = data
    .into_iter()
    .map(|(hash, mut row)| {
        let (name, targets) = seen.get(&hash).unwrap();
        if row.employee_name != *name {
            row.employee_name = name.clone();
        }
        let target = targets.get(&row.application_date).unwrap();
        if row.target != *target {
            row.target = *target;
        }
        row
    })
    .collect::<LinkedList<_>>();

let mut output_file = File::create("solicitudes-clean.csv").unwrap();
writeln!(output_file, "{headers}").unwrap();
data.iter()
    .for_each(|r| writeln!(output_file, "{r}").unwrap());

```

A continuación lee el archivo transacciones. Para cada transacción, determina la moneda local y consulta una API externa para obtener el tipo de cambio histórico a USD correspondiente a la fecha de la transacción. Añade el importe convertido en una nueva columna y guarda el resultado en el archivo de salida.

```

let file = std::fs::File::open("transacciones.csv").unwrap();
let file = std::io::BufReader::new(file);
let mut file = file.lines();

let mut headers = file.next().map(|r| r.unwrap()).unwrap();
headers.push_str("Importe USD");
let mut output_file = File::create("transacciones-clean.csv").unwrap();
writeln!(output_file, "{headers}").unwrap();

file

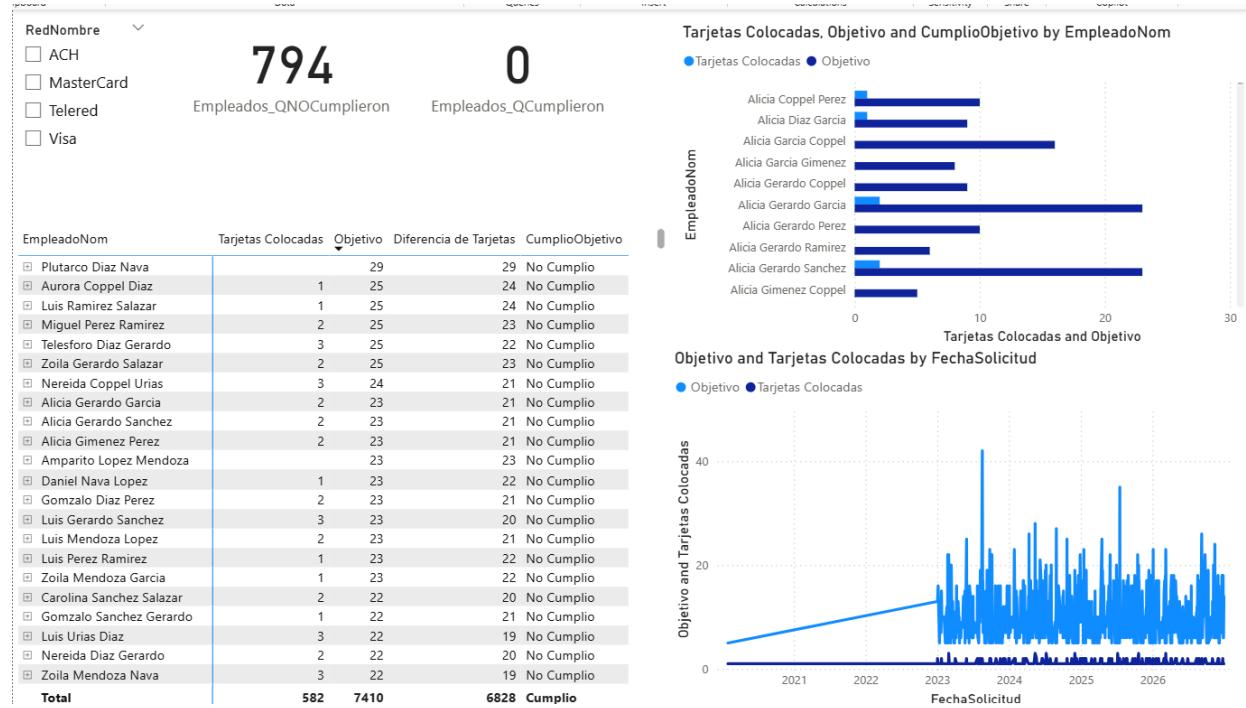
```

```
.map(Result)::unwrap)
.map(|row| Transaction::from_str(&row))
.map(|mut tran| {
    let currency_code = match tran.country.as_str() {
        "Mexico" => "MXN",
        "Japon" => "JPY",
        "Brasil" => "BRL",
        "Francia" => "EUR",
        "USA" => "USD",
        _ => unreachable!(),
    };
    let dolars = match currency_code {
        "USD" => tran.import,
        code => {
            let year = tran.date.year();
            let month = tran.date.month();
            let day = tran.date.day();

            let url = format!("https://v6.exchangerate-api.com/v6/b52d3246c5
b48151bae8546a/history/{code}/{year}/{month}/{day}");
            let currency: Currency =
reqwest::blocking::get(url).unwrap().json().unwrap();
            currency.conversion_rates.get(code).unwrap() * tran.import
        }
    };
    tran.usd = dolars;
    tran
})
.for_each(|tran| writeln!(output_file, "{tran}").unwrap());
```

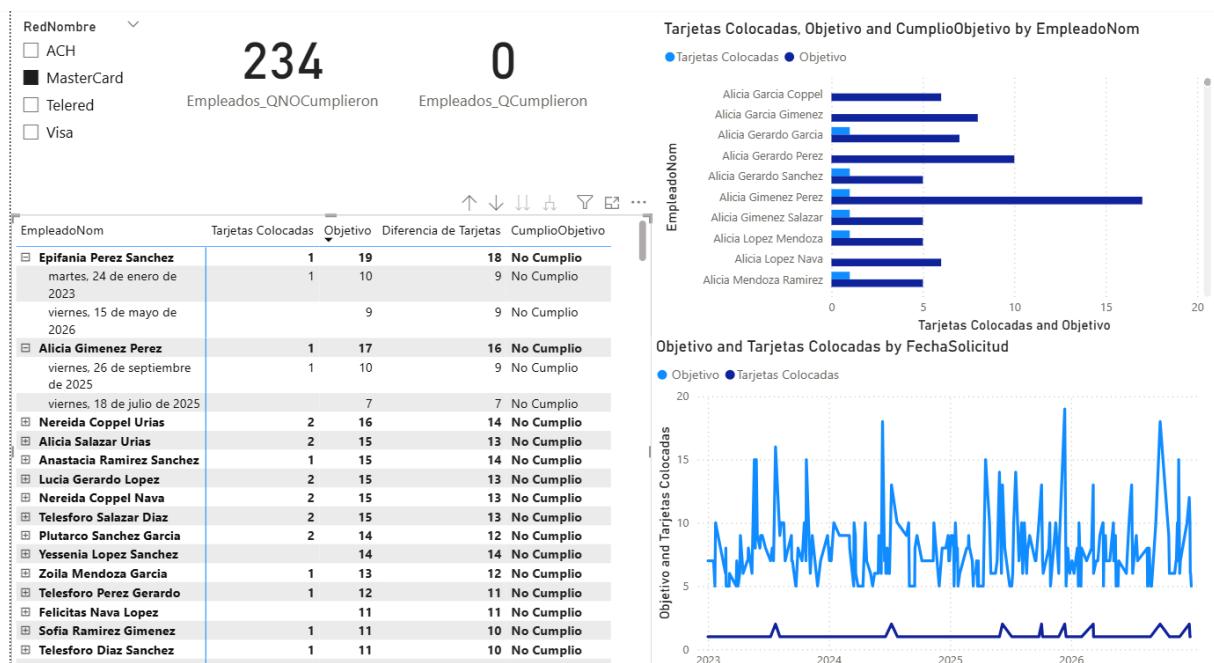
Dashboards

El primer dashboard detalla el desempeño de los empleados en relación con sus objetivos de colocación de tarjetas. Los indicadores clave de rendimiento revelan los empleados que no alcanzaron su meta diaria y los empleados que sí cumplieron su objetivo.



Asimismo, para facilitar un análisis dinámico e interactivo, se ha incorporado un slicer que permite filtrar los datos de los empleados con base en la red de tarjetas asignada. A modo de ejemplo, al seleccionar la red "MasterCard", la visualización se actualiza para mostrar exclusivamente la información correspondiente a los empleados responsables de la colocación de tarjetas de dicha red.

Proyecto de almacenes de datos

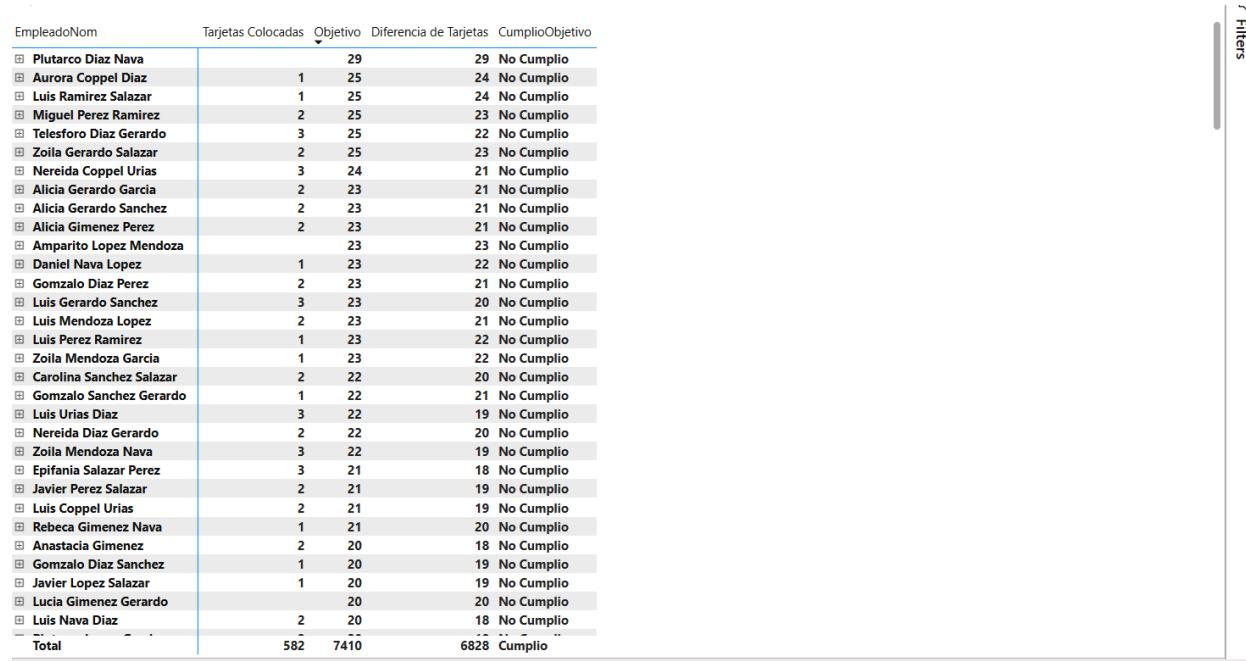


En la sección inferior, se presenta una matriz que desglosa la información a un nivel granular. Esta tabla detalla el día asignado para la colocación, el total de tarjetas colocadas, el objetivo diario estipulado y la desviación resultante con respecto a la meta.

EmpleadoNom	Tarjetas Colocadas	Objetivo	Diferencia de Tarjetas	CumplioObjetivo	Filters
Plutarco Diaz Nava	29	29	0	No Cumplio	
lunes, 26 de mayo de 2025	8	8	0	No Cumplio	
viernes, 25 de septiembre de 2026	8	8	0	No Cumplio	
sábado, 10 de febrero de 2024	7	7	0	No Cumplio	
miércoles, 13 de mayo de 2026	6	6	0	No Cumplio	
Aurora Coppel Diaz	1	25	-24	No Cumplio	
domingo, 01 de febrero de 2026	1	10	-9	No Cumplio	
miércoles, 26 de febrero de 2025	9	9	0	No Cumplio	
martes, 27 de junio de 2026	6	6	0	No Cumplio	
Luis Ramirez Salazar	1	25	-24	No Cumplio	
viernes, 18 de abril de 2026	10	10	0	No Cumplio	
jueves, 03 de diciembre de 2026	10	10	0	No Cumplio	
viernes, 20 de noviembre de 2026	1	5	-4	No Cumplio	
Miguel Perez Ramirez	2	25	-23	No Cumplio	
Telesforo Diaz Gerardo	3	25	-22	No Cumplio	
Zoila Gerardo Salazar	2	25	-23	No Cumplio	
Nereida Coppel Urias	3	24	-21	No Cumplio	
Alicia Gerardo Garcia	2	23	-21	No Cumplio	
Alicia Gerardo Sanchez	2	23	-21	No Cumplio	
Alicia Gimenez Perez	2	23	-21	No Cumplio	
Amparito Lopez Mendoza	23	23	0	No Cumplio	
Daniel Nava Lopez	1	23	-22	No Cumplio	
Gonzalo Diaz Perez	2	23	-21	No Cumplio	
Luis Gerardo Sanchez	3	23	-20	No Cumplio	
Luis Mendoza Lopez	2	23	-21	No Cumplio	

Proyecto de almacenes de datos

El dashboard presenta un análisis detallado del desempeño individual de cada empleado. La vista principal consolida las métricas clave, incluyendo el nombre del colaborador, el volumen total de tarjetas colocadas, el objetivo acumulado y la desviación neta resultante. Adicionalmente, el panel incorpora un indicador de estado que informa textualmente si el empleado “Sí cumplió” o “No cumplió” con su meta agregada.



EmpleadoNom	Tarjetas Colocadas	Objetivo	Diferencia de Tarjetas	CumplioObjetivo
Plutarco Diaz Nava	29	29	No Cumplio	
Aurora Coppel Diaz	1	25	24	No Cumplio
Luis Ramirez Salazar	1	25	24	No Cumplio
Miguel Perez Ramirez	2	25	23	No Cumplio
Telesforo Diaz Gerardo	3	25	22	No Cumplio
Zoila Gerardo Salazar	2	25	23	No Cumplio
Nereida Coppel Urias	3	24	21	No Cumplio
Alicia Gerardo Garcia	2	23	21	No Cumplio
Alicia Gerardo Sanchez	2	23	21	No Cumplio
Alicia Gimenez Perez	2	23	21	No Cumplio
Amparito Lopez Mendoza	23	23	No Cumplio	
Daniel Nava Lopez	1	23	22	No Cumplio
Gonzalo Diaz Perez	2	23	21	No Cumplio
Luis Gerardo Sanchez	3	23	20	No Cumplio
Luis Mendoza Lopez	2	23	21	No Cumplio
Luis Perez Ramirez	1	23	22	No Cumplio
Zoila Mendoza Garcia	1	23	22	No Cumplio
Carolina Sanchez Salazar	2	22	20	No Cumplio
Gonzalo Sanchez Gerardo	1	22	21	No Cumplio
Luis Urias Diaz	3	22	19	No Cumplio
Nereida Diaz Gerardo	2	22	20	No Cumplio
Zoila Mendoza Nava	3	22	19	No Cumplio
Epifania Salazar Perez	3	21	18	No Cumplio
Javier Perez Salazar	2	21	19	No Cumplio
Luis Coppel Urias	2	21	19	No Cumplio
Rebeca Gimenez Nava	1	21	20	No Cumplio
Anastacia Gimenez	2	20	18	No Cumplio
Gonzalo Diaz Sanchez	1	20	19	No Cumplio
Javier Lopez Salazar	1	20	19	No Cumplio
Lucia Gimenez Gerardo	20	20	No Cumplio	
Luis Nava Diaz	2	20	18	No Cumplio
Total	582	7410	6828	Cumplio

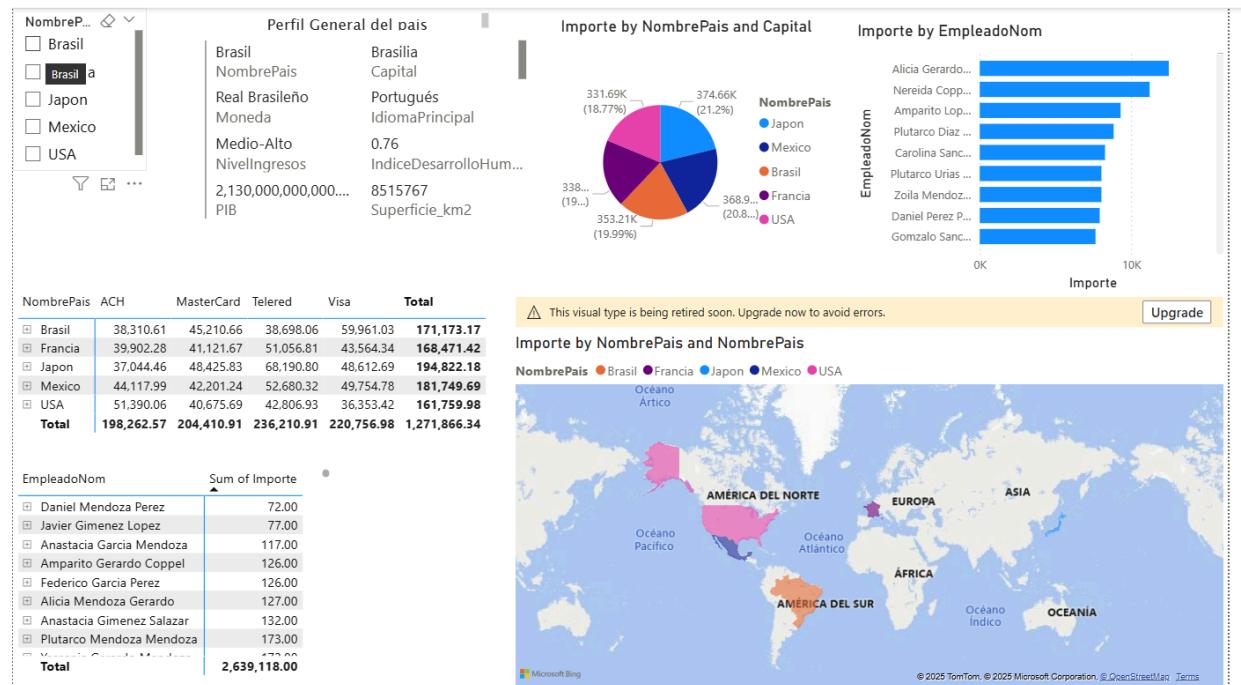
El segundo dashboard ofrece un análisis detallado de los importes generados por país, permitiendo cuantificar su contribución al volumen total.

El panel principal expone el perfil demográfico y económico de cada nación, incluyendo datos relevantes como su capital, idioma, nivel de ingresos, PIB y superficie, con el fin de contextualizar los resultados financieros. Una tabla en la sección inferior desglosa los importes totales por red de tarjeta, lo cual facilita un análisis comparativo entre países y redes.

Un gráfico circular complementa esta vista, ilustrando la participación porcentual de cada país sobre el importe total. Paralelamente, un gráfico de barras identifica a los empleados con la mayor contribución económica.

Finalmente, un mapa geográfico interactivo permite visualizar la distribución de los importes a escala global, brindando una perspectiva clara sobre las regiones con mayor desempeño financiero. Este dashboard es una herramienta clave para la evaluación del rendimiento por país, la detección de oportunidades de crecimiento y la optimización de estrategias comerciales a nivel internacional.

Proyecto de almacenes de datos



El dashboard desglosa los importes generados, segmentándolos por país y red de tarjeta, e indicando los totales correspondientes.

La herramienta permite una segmentación secundaria; al seleccionar un país específico, los datos se desagregan por semestre. Esto facilita el análisis de la evolución temporal del rendimiento. Al desplegar la información semestral, se exponen los importes correspondientes a cada mes dentro del período seleccionado.

Si se requiere un nivel de detalle superior, el dashboard permite una profundización adicional, mostrando las fechas exactas en las que se registraron los importes y el monto específico de cada uno.

Dicha estructura posibilita la identificación precisa de los montos generados por cada combinación de país y red, facilitando una comparativa directa del desempeño económico y la detección de las regiones o redes que aportan mayor valor a la organización.

NombrePaís	ACH	MasterCard	Telered	Visa	Total
Brasil	38,310.61	45,210.66	38,698.06	59,961.03	171,173.17
1	10,580.36	20,981.50	18,108.57	23,707.21	70,735.28
April	556.26	4,647.87	3,505.32	1,758.91	10,468.36
February	3,598.03	1,825.55	5,586.32	6,414.30	17,424.20
January	2,016.30	300.70	3,712.75	5,937.94	10,451.75
June	2,728.65	6,040.90	2,444.25	3,032.40	13,119.80
March	824.63	3,764.71	1,997.05	1,507.41	8,093.80
May	856.49	4,401.77	862.88	5,056.25	11,177.38
2	27,730.25	24,229.16	20,589.49	36,253.82	100,437.89
August	4,887.12	4,780.89	1,711.76	7,309.65	18,254.26
December	5,402.76	7,083.11	6,686.12	4,782.78	20,210.06
July	3,481.10	1,465.85	1,396.58	5,107.20	11,450.73
November	3,171.93	1,559.48	3,120.56	5,989.90	13,841.87
October	6,271.17	3,367.55	5,276.19	7,565.42	21,485.43
September	4,516.17	5,972.28	2,398.28	5,498.87	15,195.53
Francia	39,902.28	41,121.67	51,056.81	43,564.34	168,471.42
1	22,498.79	24,293.66	26,341.16	21,750.88	91,777.92
April	3,251.66	3,432.18	2,646.79	2,072.81	10,902.28
February	866.20	951.44	9,329.43	8,159.75	19,306.82
January	5,335.49	8,541.38	4,954.84	17,262.42	
June	1,756.47	2,109.57	1,423.14	3,606.52	8,895.70
March	2,517.92	5,415.10	2,297.37	3,767.21	12,961.49
May	8,771.05	3,843.98	5,689.59	4,144.59	22,449.21
2	17,403.49	16,828.01	24,715.65	21,813.46	76,693.50
August	3,332.55	4,463.91	5,292.87	3,341.90	15,538.81
December	6,712.38	2,503.32	4,242.06	2,162.44	15,620.20
July	1,101.64	2,518.23	1,214.33	1,715.24	6,549.44
November	4,345.24	2,266.74	3,014.94	2,678.17	12,305.09
October	919.89	5,516.62	3,196.57	7,450.35	
September	1,911.69	4,155.93	5,434.83	8,719.13	19,229.62
Japon	37,044.46	48,425.83	68,190.80	48,612.69	194,822.18
Total	198,262.57	204,410.91	236,210.91	220,756.98	1,271,866.34

Conclusiones

Abitia Burgos Julio Alejandro:

En este proyecto entre las cosas que mas me llevo es el uso de la herramienta Power BI ya que aprendí a manejarla de una buena manera, al igual a la hora de manejar el lenguaje DAX que es fácil de usarlo y te ayuda hacer un mejor trabajo a la hora de crear medidas y tablas. Al igual uno de los aprendizajes que me llevo de este proyecto es la importancia de hacer la limpieza de los datos, ya que una vez que la información está limpia la puedes manipular como tú quieras y a la hora de usarla puede representar muchas cosas.

Ramos Matunaga Raúl Alejandro:

Este proyecto fue mi primer acercamiento a Power BI y al lenguaje de expresiones DAX, lo que me permitió iniciarme en el área de ciencia de datos, algo que desconocía completamente y que me hizo ver su importancia, así como conocer las capacidades de Power BI, la facilidad para la creación de complejos dashboards y su importancia a la hora de llevar a cabo la toma de decisiones con base en datos.

Yair Torres Lizárraga:

Este proyecto me permitió ver la importancia de la limpieza de los datos y cómo esta puede tener un gran impacto en la generación de resultados confiables. Aprendí a utilizar la herramienta Power BI, comprendiendo su integración con el lenguaje DAX, el cual facilitó la creación de

medidas y columnas calculadas.

Además, pude notar las ventajas que ofrece DAX, al permitir reducir la cantidad de código necesario y brindar soluciones más limpias, dinámicas y reutilizables para el análisis de la información.