



**UNIVERSIDAD
DE ANTIOQUIA**

**AUTOMATIZACIONES TARJETA VISA
PREPAGO BANCOLOMBIA**

Autor
Jeferson Velásquez Ortiz

Universidad de Antioquia
Facultad de Ingeniería
Medellín, Colombia
2021



Automatizaciones tarjeta visa prepago Bancolombia

Jeferson Velásquez Ortiz

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:

Ingeniero de sistemas

Asesores:

Javier Fernando Botía Valderrama
Doctor en Ingeniería Electrónica
Docente Departamento Ingeniería de sistemas

Jair Eduardo Liévano Bustamante
Gerencia de Servicios en Medios de Pago
Jefe sección Apoyo al tarjetahabiente

Leonardo Pérez Flórez
Sección Apoyo al tarjetahabiente
Analista II

Universidad de Antioquia

Facultad de Ingeniería

Medellín, Colombia

2021.

Tabla de contenido

RESUMEN	3
PALABRAS CLAVES	3
INTRODUCCIÓN	4
OBJETIVOS	5
▪ <i>Objetivo General</i>	<i>5</i>
▪ <i>Objetivos Específicos</i>	<i>5</i>
MARCO TEÓRICO	5
○ SQL.....	6
○ PYTHON	6
○ PANDAS	6
○ DJANGO	6
○ HTML	7
○ CSS	7
○ JAVASCRIPT	7
METODOLOGÍA	8
○ CONSULTAS SQL.....	8
○ WEB APP	9
○ CONEXIÓN A BASE DE DATOS	9
○ AUTENTICACIÓN WEB APP	10
○ VISUALIZACIÓN (FRONTEND).....	11
○ DESPLIEGUE.....	12
RESULTADOS Y ANÁLISIS	12
CONCLUSIONES	16
REFERENCIAS	17

Automatizaciones tarjeta visa prepago Bancolombia

Resumen

En el área de visa prepago, los auxiliares que acceden al historial de transacciones de las tarjetas, por lo general, realizan las consultas en la base de datos en lenguaje SQL debido a la falta de algún aplicativo para este tipo de tarea. En muchos casos, los auxiliares no tienen ningún tipo de formación en Ingeniería de Sistemas o áreas afines, por lo cual, utilizan las consultas ya preestablecidas con poco margen para su correcta modificación. Esta limitante puede ocasionar algunos errores de búsqueda para detectar anomalías en transacciones bancarias o hacer consultas especializadas con algún margen de error. Por consiguiente, los auxiliares realizan labores repetitivas que no generan un avance significativo en el análisis de las transacciones y, además, influye en frustraciones por parte del auxiliar que ocasionan disminución en su desempeño laboral. A causa de estas problemáticas, se propone el desarrollo de un aplicativo web que, mediante el ingreso de los datos a consultar, evite preocupar al auxiliar por intentar entender el código de la consulta para su modificación. De esta manera, la aplicación permite consultar las transacciones de VISA Prepago, que a través del ingreso del campo que se desea consultar y la activación de algún filtro de búsqueda de interés para el auxiliar, en pocos segundos se obtienen los resultados esperados con un par de clics. El desarrollo de la aplicación se centró en darle al auxiliar un ambiente amigable y organizado para facilitar la búsqueda de transacciones bancarias VISA Prepago, dando la posibilidad de realizar la exportación de los resultados a diferentes formatos para su análisis, manipulación o presentación a los clientes.

Palabras claves

SQL, Python, Django, framework, IIS, HTML, CSS, JavaScript, Visa Prepago, Base de datos, aplicativo web, wFastCgi.

Introducción

El producto VISA Prepago es una solución a nivel empresarial que permite realizar pagos a terceros de una forma segura y disponer del dinero de caja menor de las empresas de una forma más organizada. Este producto es relativamente nuevo en el mercado y cuenta actualmente con cerca de 100.000 tarjetas (Tarjeta prepago Bancolombia Empresas, 2021). Este producto es totalmente independiente de los demás ofrecidos por el banco en el tema de tarjetas de crédito, por lo cual cuenta con procesos autónomos que funcionan de manera diferente y con una base de datos independiente.

En la actualidad existe la necesidad de realizar varias consultas a la base de datos para la realización de comprobaciones y reportes, tanto a nivel de clientes como a nivel administrativo. Estas consultas son realizadas en lenguaje SQL y el problema radica en que estas operaciones son ejecutadas por personas sin conocimientos en programación, dando pie a posibles errores a la hora de obtener la información y existiendo un alto riesgo de errores humanos, conllevando esto al posible deterioro de la imagen de la marca.

Como se puede apreciar estos procesos rutinarios son realizados día a día de una manera manual, principalmente por los auxiliares del área de VISA Prepago, generando así un desgaste de tiempo y además con una alta probabilidad de ocurrencia de errores, lo que podría conllevar a la pérdida de productividad para Bancolombia. Todo esto podría evitarse si se tuviera una herramienta que ejecute todas estas consultas sin que el auxiliar tenga que manipular el código SQL.

La idea con la ejecución del semestre de industria es realizar un análisis de esos procesos repetitivos para poder crear procesos automáticos los cuales sean ejecutados de manera programada, según sea la necesidad. Además, revisar las consultas SQL que actualmente se ejecutan y verificar su correcto funcionamiento y si es necesario realizar ajustes en ellas. Dichas consultas serán solamente de modo SELECT, por seguridad no se implementarán consultas que puedan modificar la base de datos. Asimismo, se espera que los procesos anteriormente mencionados puedan ser ejecutados desde una herramienta más amigable con posibles ajustes en los campos de las consultas, por ejemplo, a través de un portal web, el cual será realizado en Python y utilizando el **Framework**

Django. Adicionalmente, se agregará una sección para revisar algunos controles que se ejecutarán con solo un clic.

En el presente trabajo, se mostrará la forma en que se abordó dicho problema, así como la metodología utilizada e igualmente las tecnologías y herramientas implementadas en la solución.

Objetivos

▪ **Objetivo General**

Desarrollar un software aplicativo que permita una mayor eficiencia en los procesos realizados sobre la base de datos del producto visa prepago.

▪ **Objetivos Específicos**

- Analizar las consultas SQL y los procesos repetitivos que se encuentran en el área de visa prepago.
- Proponer consultas SQL más acordes a las necesidades de los procesos y su implementación en una página web.
- Evaluar el funcionamiento de la solución planteada para corroborar su desempeño de acuerdo con las necesidades del producto visa prepago.

Marco Teórico

Las tecnologías de la información han impulsado de gran manera las organizaciones en los últimos años, brindándole a estas la posibilidad de automatizar los procesos y mejorar los resultados, entre estos le permite realizar en pocos minutos acciones mecánicas o rutinarias que tomaban mucho tiempo (Cano-Pita, 2018).

En el presente caso se busca mejorar los procesos realizados en el área de visa prepago y para ello se implementará una solución basada en Python utilizando el *Framework Django*, por medio del

cual se realizarán consultas SQL a la base de datos y estos resultados se presentarán de una forma más entendible para el usuario final (auxiliares).

- ***SQL***

Sus siglas en español se refieren a lenguaje de consulta estructurada (*Structured Query Language*) es un lenguaje de programación que fue diseñado para administrar y consultar información de sistemas de gestión de bases de datos (SGBD) (Morteo, Francisco y Bocalandro, Nicolás, 2004). Las sentencias SQL son utilizadas para realizar tareas tales como crear, actualizar, borrar o recuperar datos de una base de datos. Entre los principales administradores de bases de datos relacionales que usan SQL están: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc.

- ***Python***

Python es un lenguaje de programación Open Source lo que nos permite la utilización de este de manera libre sin restricciones en su distribución (Challenger-Pérez, Díaz-Ricardo, & Becerra-García, 2014). Se ha decidido la utilización de Python gracias a su gran madurez y soporte, además su ya utilización en la compañía aporta un soporte interno, también gracias a que cuenta con varios servidores de aplicaciones y Frameworks. “La web se ha convertido hoy en el centro de la vida digital del planeta. Un lenguaje que no cuente con el debido soporte técnico de los distintos servidores de aplicaciones web puede considerarse un lenguaje obsoleto” (Challenger-Pérez, Díaz-Ricardo, & Becerra-García, 2014, pág. 9). Además, Python es el lenguaje por excelencia para el desarrollo de procesos de aprendizaje computacional (machine learning) (euskadinnova, 2021) y con esto se tiene la de agregarle en un futuro funcionalidades que nos permitan el análisis de los datos mediante esta rama de la inteligencia artificial.

- ***Pandas***

Pandas es una biblioteca de software que hace parte de las extensiones de Numpy, y es utilizada para la manipulación y el análisis de datos con el lenguaje de programación Python.

- ***Django***

Django es un framework web de alto nivel utilizado para el desarrollo rápido de aplicaciones web seguras y mantenibles. Es un framework gratuito y de código abierto el cual cuenta con una comunidad activa y una gran documentación (MDN contributors, 2021)

- ***HTML***

HTML (HyperText Markup Language) es un lenguaje de marcado de hipertexto con el cual se escriben la mayoría de las páginas web. Es un documento que combina textos, imágenes y otras características multimedia. Dicho documento puede ser accedido, interpretado y renderizado por un navegador. (Eguiluz, Introducción a XHTML, 2008).

Estos documentos se componen de una serie de etiquetas predefinidas que añaden estructura, jerarquía y comportamiento a la página, pudiendo organizar la información en bloques semánticos y añadiendo enlaces entre las distintas secciones y documentos.

La versión actual de HTML es la 5 y es utilizado extensivamente para el desarrollo de páginas webs que se componen usualmente de un conjunto de documentos HTML, imágenes, hojas de estilo y *scripts*.

- ***CSS***

Hojas de Estilo en Cascada (Cascading Style Sheets) o CSS es un lenguaje utilizado para describir la presentación de documentos HTML. Mediante CSS se describe cómo debe ser renderizado el elemento estructurado en la pantalla.

En cada hoja de estilo se definen las reglas que permiten establecer los valores de los atributos de los distintos elementos o secciones de una página, por ejemplo, establecer el color y tamaño del texto o imagen. Cada regla está definida por un selector (por ejemplo, ID) y un conjunto de pares atributo-valor, los cuales se aplicarán sobre las selecciones encontradas en el documento (Eguiluz, 2008).

- ***JavaScript***

JavaScript es el lenguaje de scripting más difundido y utilizado para el desarrollo web. Gracias a este lenguaje se reduce el número de llamadas al servidor, por ejemplo, cuando se necesitaban validar datos de un formulario (Eguiluz, Introducción a JavaScript, 2009).

En la actualidad es utilizado extensamente para mejorar la experiencia de los usuarios a través de la incorporación de componentes personalizados, actualizaciones de contenido sin recargar la página (AJAX), entre otras características. Además, es un lenguaje que puede ser utilizado tanto en el cliente web (Front-End) como en lado del servidor (Back-end) con su respectiva conexión a

una base de datos y es usado también para el desarrollo de aplicaciones tanto para dispositivos móviles y de escritorio (Gallardo, 2017)

Metodología

En la ilustración 1, se ilustra la metodología seguida para el desarrollo del aplicativo web:

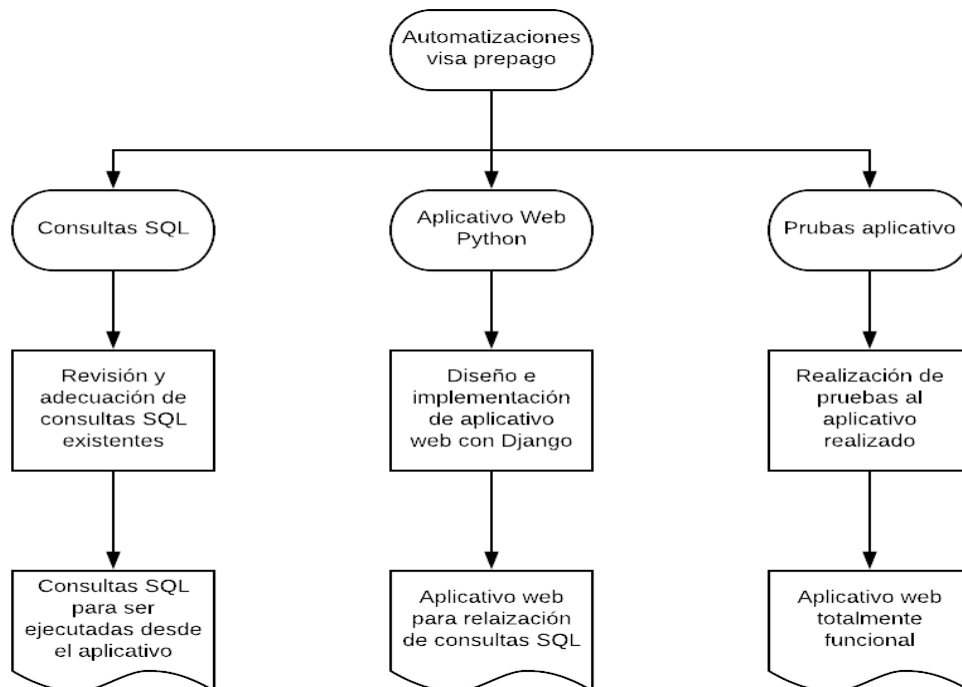


Ilustración 1 Mapa desarrollo aplicativo web. [Fuente: Autor]

○ **Consultas SQL**

Inicialmente se procedió a realizar reuniones con la persona encargada de las consultas SQL para hacer una validación de dichas consultas, así como para crear nuevas consultas necesarias para el área.

Una vez obtenidas las consultas a utilizar, se procedió a probarlas por un periodo de 2 semanas, en dicho periodo se realizaron algunos ajustes gracias a la retroalimentación de los auxiliares del área de VISA Prepago.

Una vez listas las consultas SQL que se utilizarán en el aplicativo, se procede a la implementación de este. Para esto, por políticas del banco se opta por utilizar el lenguaje de programación Python, y para la solución como aplicativo web se define la utilización del framework Django, ya que este es un entorno de trabajo muy amigable para las soluciones de aplicativos webs.

- ***Web App***

Este proyecto está desarrollado en el lenguaje de programación Python, esto es debido a que es el lenguaje que el banco está utilizando en la actualidad para el desarrollo de aplicaciones. Además, se utilizó el framework Django, el cual facilita la creación de aplicativos web, ya que gracias a este es posible el desarrollo tanto del Backend como del FrontEnd desde él mismo y hace que el enrutamiento y el manejo de las URLs sea bastante sencillo.

Para el presente proyecto se creó un aplicativo con Django establecido en un módulo, el cual se crearon las respectivas vistas para el manejo de las consultas y su posterior visualización en la página web.

- ***Conexión a Base de datos***

El primer punto que se abordó fue la conexión del aplicativo con la base de datos del banco, la cual está sobre un **IMB i AS400**. Los drivers proporcionados por IBM para dicha conexión con Django, a la fecha de la implementación, no se encontraban totalmente funcionales. En su repositorio oficial en la página GitHub se estaba a la espera de la corrección de algunos errores relacionados con este, ya que a la hora de la utilización se generaban errores al conectar con la base de datos (Gibhub, 2021).

Debido a la imposibilidad de utilizar los drivers proporcionados por IBM se optó por utilizar ODBC, que es un estándar para realizar el acceso a las bases de datos sin importar el tipo de sistema de gestión de bases de datos (Universidad de Valencia, 2021).

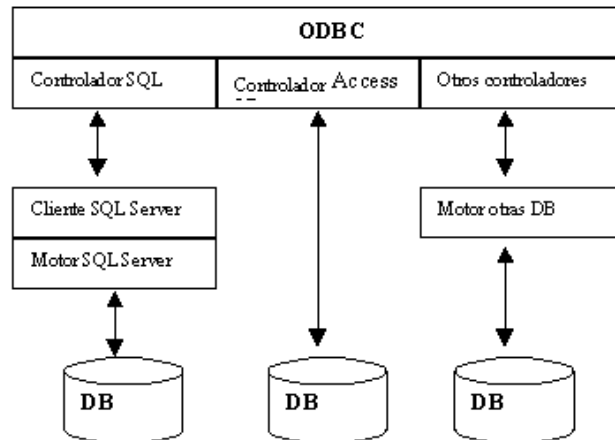


Ilustración 2. Flujo trabajo driver ODBC [Fuente: (Universidad de Valencia, 2021)]

Para utilizar ODBC a través de Python fue necesario utilizar el módulo **pyodbc** el cual también es considerado como un controlador SQL para Python.

Una vez establecida la conexión con la base de datos, se establece la necesidad de usar herramientas para la manipulación de las tablas obtenidas en las consultas y su correcta visualización en el aplicativo. Para ello, se utilizó una librería de Python llamada *pandas*, ampliamente conocida en el ámbito de la manipulación de datos. Mediante la librería *pandas* se hizo la conversión de las respuestas obtenidas por las consultas, que vienen dadas en texto plano, a un DataFrame. Un Data Frame es una estructura de datos bidimensional con filas y columnas el cual nos permite realizar operaciones sobre esta y nos facilita la adecuada manipulación para su posterior visualización del aplicativo.

○ ***Autenticación Web App***

Uno de los requisitos expresados por los auxiliares del área de VISA Prepago fue que la autenticación en el aplicativo se hiciera a través de los usuarios existentes. Esto también es considerado un requisito no funcional ya que las consultas en la base de datos se deben realizar a través de los usuarios ya establecidos, que cuentan con los permisos necesarios para el acceso a las tablas consultadas. Para esto se hizo uso de los Backends de autenticación que ofrece Django,

estos extienden las propiedades y métodos de la autenticación por defecto de Django, para así poder utilizar otros métodos de autenticación que funcionen en los aplicativos (Django Software Foundation, 2021). Para el uso de este Backend, se utilizó el módulo *pyodbc*, anteriormente mencionado, este fue el encargado de la comunicación con la base de datos para verificar la autenticación de los usuarios.

Una vez confirmado el usuario y contraseña, la persona identificada en el sistema tiene acceso a las consultas sobre la base de datos. Además, para que el usuario pueda seguir identificado en el aplicativo después de cerrar el navegador, se utilizaron las sessions que provee Django, estas nos permiten almacenar y posteriormente recuperar la sesión del usuario identificado. Estos datos son almacenados en forma de cookies en el navegador y por seguridad se guarda un ID de la sesión y no los datos de usuario y contraseña (Django Software Foundation, 2021).

○ ***Visualización (FrontEnd)***

Para la parte de la visualización en el aplicativo (FrontEnd) se hizo uso de *HTML*, *CSS*, *Bootstrap*, *JavaScript*, *AJAX* y *jQuery*. Se utilizó la versión 5 de *Bootstrap* que es la más reciente, además para la validación de campos en los formularios se utilizó la librería *JQuery-Validate* y se complementa con código en *JavaScript*. Además, se realizaron algunas modificaciones a la página por medio de *CSS* y se utilizaron algunos logos, imágenes y estilos proporcionados por el banco.

Para la parte de la visualización de las consultas, se tomó la respuesta obtenida en un *DataFrame* y se realizó la conversión a formato *JSON*, esto último para su correcto envío desde la vista de Python hacía el archivo *HTML*. Una vez el archivo en formato *JSON* es recibido por el navegador a través del *HTML*, éste renderiza la información en una tabla estilizada que cuenta con filtros por columnas y además cuenta con botones para descargar en diferentes formatos (*Excel*, *CSV*, *pdf*) además también cuenta con un botón para copiar todas las filas filtradas y otro para imprimir. Para estas últimas funcionalidades se utilizó el complemento *DataTable*, el cual está basado en *JavaScript* y ofrece estas funcionalidades antes mencionadas (Pypi, 2021).

○ *Despliegue*

Para el despliegue de la aplicación se utilizó un equipo con sistema operativo Windows 10, en el cual se habilitó el *Internet Information Services* (IIS) con el que se realizó la publicación del aplicativo web. Para ello, fue necesario la utilización de un módulo de Python llamado Wfastcgi, el cual es el encargado de la conexión entre los aplicativos Python y el IIS. Dicho aplicativo fue desplegado solo a nivel local de la red de Bancolombia y por ende solo puede ser accedido a través de la red corporativa del banco.

Resultados y análisis

El aplicativo web realizado fue diseñado con el manual de marca del banco, utilizando los colores y los logos proporcionados por ellos, a continuación, se muestran algunas capturas de pantalla del aplicativo en funcionamiento:

La siguiente imagen muestra la pantalla de inicio de la aplicación, la cual es el *log in*, solo se puede acceder a las funcionalidades del aplicativo iniciando sesión en el sitio.



Ilustración 3. Pantalla de Log in [Fuente: Bancolombia]

Una vez se inicia sesión correctamente, se puede observar la pantalla con las opciones que actualmente cuenta el aplicativo, en la pestaña de transacciones se puede realizar la búsqueda de las transacciones de tres formas diferentes, por medio del número de cuenta, el número del NIT de la empresa o el número de tarjeta de crédito. Además, estas opciones cuentan con la posibilidad de filtrar por fecha inicial, fecha final o por el logo de la tarjeta de crédito (logo se refiere al tipo de producto visa prepago, ya que la marca cuenta con 6 productos diferentes). También se cuenta con la opción de ponerle límite a la consulta SQL.

Consultas Transacciones (Actual) Controles Usuario: njefvelas Logout

Por Cuenta Por NIT Por Tarjeta

Ingrese la cuenta

Ingresar fecha inicial

Ingresar fecha final

Elegir Logo

Limit 1500

Realizar Consulta

Ilustración 4. Página Consultas [Fuente: Bancolombia]

En la siguiente imagen, podemos visualizar la pantalla de resultados, en donde se especifica el número de resultados que trajo la consulta y el número que se están mostrando actualmente, en el caso que se haya utilizado alguno de los filtros disponibles.

Se puede observar como por cada uno de los campos de la consulta se puede realizar el filtro, así como un filtro general para buscar en todos los campos.

En la parte superior izquierda se pueden observar los botones que nos permiten descargar los resultados en los diferentes formatos. Además, al final de la tabla observamos las opciones de

paginación, elegir resultados por página y el número de páginas en los que se muestran los resultados.



Ilustración 5. Página resultados [Fuente: Bancolombia]

Después de presentar el aplicativo web a los usuarios finales, quienes son los auxiliares que necesitan consultar las transacciones de las tarjetas visa prepago, se contó con 12 días de pruebas en los que nos compartieron la retroalimentación y los errores presentados durante su utilización. En estos 12 días se recibieron los comentarios de 5 personas que fueron las encargadas de realizar las pruebas. El principal error o mal funcionamiento detectado por estas personas fue la exportación de los resultados a el formato Excel, el cual presentaba algunos datos de forma errónea, principalmente el valor de la transacción en la mayoría de los casos le agregaba un cero de más. Se procedió a realizar la validación y se encontró que el error ese presentaba por el tipo de formato en el que se obtenían los resultados de la base de datos era diferente al formato utilizado por DataTables a la hora de realizar la exportación a Excel, la base de datos utiliza la coma (,) como separador decimal y Excel utiliza la coma (,) como separador de miles. Se logró solucionar este problema realizando el ajuste de los datos recibidos por la base de datos antes de la exportación a Excel.

Otra de las retroalimentaciones recibidas fueron las pantallas de error del aplicativo, las cuales no eran muy dicientes, solo se podía observar el número de error presentado, por ejemplo, error 400 o error 500, y dicha información no era muy explicativa acerca de lo que había ocurrido mal. Debido a esto se implementaron varias pantallas de error, siguiendo el manual de marca del banco las cuales se presentan en las siguientes imágenes. Como podemos ver ya se puede leer un poco

acerca de lo que salió mal a la hora de utilizar el aplicativo y se obtiene de una manera más amigable para el usuario final.

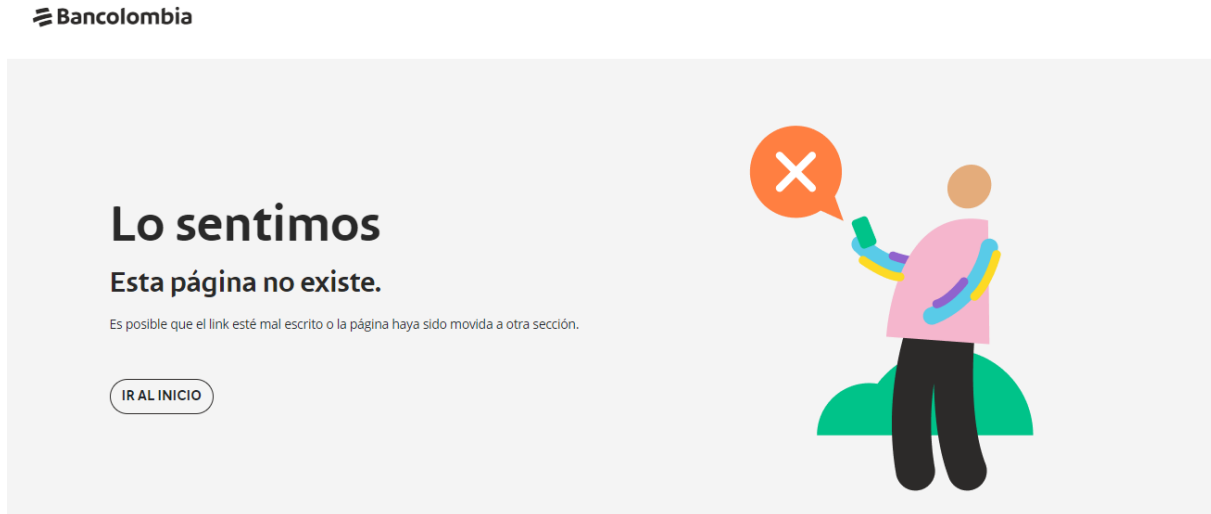


Ilustración 6. Página de error 1. [Fuente: Bancolombia]

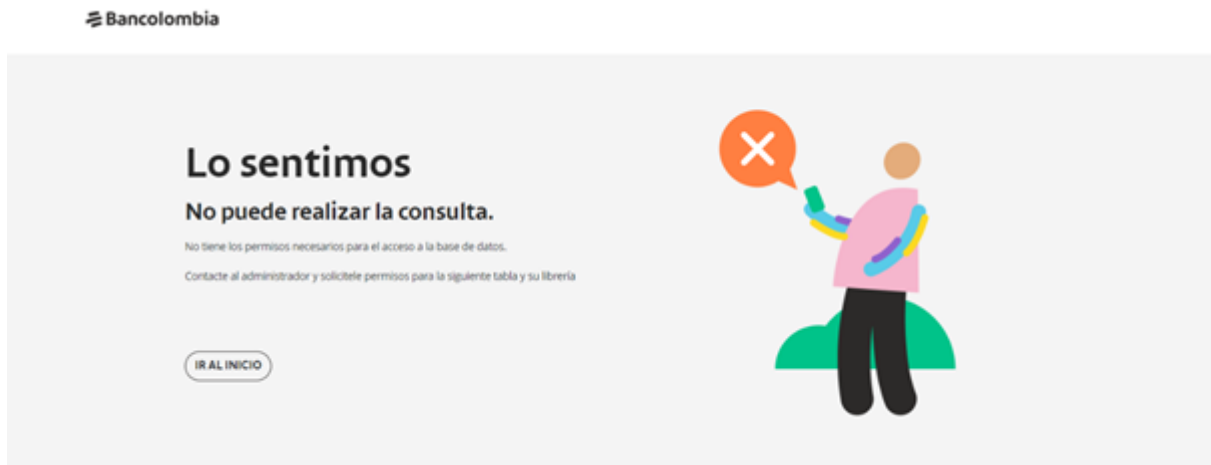


Ilustración 7. Página de error2. [Fuente: Bancolombia]

Conclusiones

- Se pudo constatar que las consultas SQL utilizadas en el área estaban bien estructuradas y funcionaban muy bien, solamente se hicieron algunas adecuaciones para mejorar la salida final de la información, para que fuera mucho más entendible (nombre de campos).
- Una vez obtenidas las consultas que satisfacían las demandas de los auxiliares, se desarrolló un aplicativo web el cual mejoró la eficiencia de los auxiliares a la hora de obtener las transacciones de las tarjetas de los clientes de visa prepago. Con lo cual pueden ocupar más espacio en el análisis y obtención de alguna irregularidad en los movimientos que en la obtención de estos.
- Se pudo constatar que el aplicativo web funciona con normalidad incluso con 5 o 6 personas utilizando a la vez, que es el flujo máximo esperado en simultáneo, lo cual no indica que el aplicativo responde satisfactoriamente a las necesidades planteadas por los auxiliares.
- Gracias al uso de Python como lenguaje de programación, se deja la puerta abierta para futuras funcionalidades como por ejemplo la ejecución automática de algunos controles sobre la base de datos y su respectiva notificación.

Referencias

- Cano-Pita, G. E. (enero de 2018). Las TICs en las empresas: evolución de la tecnología y cambio. *Dominio de las Ciencias*, 4(1), 507-508.
- Challenger-Pérez, I., Díaz-Ricardo, Y., & Becerra-García, R. A. (abril-junio de 2014). El lenguaje de programación Python. *Ciencias Holguín*, 20(2), 1-13.
- Django Software Foundation. (05 de 2021). *django*project. Obtenido de <https://docs.djangoproject.com/en/3.2/topics/http/sessions/>.
- Django Software Foundation. (05 de 2021). *django*project. Obtenido de <https://docs.djangoproject.com/en/3.2/topics/auth/>
- Eguiluz, J. (2008). *Introducción a CSS*. Recuperado el 15 de mayo de 2021, de https://www.jesusda.com/docs/ebooks/introduccion_css.pdf
- Eguiluz, J. (2008). *Introducción a XHTML*. Recuperado el 15 de Mayo de 2021, de https://www.jesusda.com/docs/ebooks/introduccion_xhtml.pdf
- Eguiluz, J. (2009). *Introducción a JavaScript*. Recuperado el 5 de mayo de 2021, de https://www.jesusda.com/docs/ebooks/introduccion_javascript.pdf
- euskadinnova. (01 de Mayo de 2021). *Introducción a Python para Machine Learning (online)*. Obtenido de <https://www.spri.eus/euskadinnova/es/empresa-digitala/agenda/introduccion-python-para-machine-learning-online/16741.aspx>
- Gallardo, A. (2017). *Sistema de Administración de Contenido en la nube con integración en clientes web*.
- Github. (Febrero de 2021). *Github*. Obtenido de <https://github.com/ibmdb/python-ibmdb/issues/604>
- MDN contributors. (02 de Mayo de 2021). *developer.mozilla*. Obtenido de <https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>
- Morteo, Francisco y Bocalandro, Nicolás. (2004). *Un enfoque práctico de SQL*.
- Pypi. (05 de 2021). <https://pypi.org/project/wfastcgi/>.
- Tarjeta prepago Bancolombia Empresas. (11 de 02 de 2021). Obtenido de <https://www.grupobancolombia.com/wps/portal/empresas/productos-servicios/cash-management/pagos/tarjeta-prepago>
- Universidad de Valencia. (05 de 2021). *Guia para escribir documentos HTML*. Obtenido de <https://www.uv.es/jac/guia/gestion/gestion3.htm>