



Sistemas de información

Trabajo Fin de Grado

Sistema de agregación de información bancaria para entornos empresariales

ESCUELA POLITECNICA
SUPERIOR

Autor: Adrián Blanco Domínguez

Tutor/es: Óscar García Población

2022

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Grado en sistemas de información

Sistema de agregación de información bancaria para entornos empresariales

Autor: Adrián Blanco Domínguez

Tutor/es: Óscar García Población

TRIBUNAL

Presidente: Elena Campo Montalvo

Vocal 1º: Concepción Batanero Ochaíta

Vocal 2º: Óscar García Población

FECHA: 25 de marzo de 2022

Contenido

Resumen	5
Summary	7
Introducción	9
Estado del arte	11
Análisis de mercado	13
Marco legal	15
CANVAS	17
Análisis de la solución	21
Diagrama de casos de uso	22
Diagrama entidad-relación	26
Diseño de la solución	29
Diagrama de clase	31
Esquema de funcionamiento	35
Descripción técnica	36
Arquitectura Cliente-Servidor	43
Diagrama de red	45
Funcionamiento del sistema de scrapping	47
Implantación	49
Resultados del proceso de scrapping	53
Fichero de exportación	57
Interfaz Gráfica	63
Pantalla de Login y formulario de registro	63
Menú principal	64
Resultados	65
Relación con el grado	67
Conclusiones	69
Bibliografía	71

Resumen

Este trabajo de Fin de Grado muestra el análisis de los procesos contables que se realizan de forma manual dentro de los entornos empresariales, en concreto el volcado de datos bancarios de forma automática a los sistemas de gestión contable que utilizan las empresas. Ante esta problemática, se propone una solución automática con el fin de poder extraer información de carácter contable de las entidades bancarias, para posteriormente ser procesada y exportada de manera legible para los sistemas de información. Para realizar esta solución utilizaremos tecnologías de automatización de procesos y programación para el procesamiento de los datos

Palabras clave: Contabilidad, Automatización, ERP, Banca electrónica, Scrapping.

Summary

This Final Degree Project shows the analysis of the accounting processes that are carried out manually within business environments, specifically the transfer of bank data automatically to the accounting management systems used by companies. Faced with this problem, an automatic solution is proposed in order to be able to extract accounting information from banking entities, to later be processed and exported in a legible way for information systems. To carry out this solution we will use process automation and programming technologies for data processing

Keywords: Accounting, Automation, ERP, Electronic Banking, Scrapping.

Introducción

El interés por el tema objeto de estudio surge a partir del alzamiento de nuevas tecnologías que permiten la automatización de procesos, en este caso en particular, procesos empresariales, así como la integración de todos estos con un nuevo modelo de negocio basado en internet. El usuario de servicios se enfrenta, cualquiera que sea su nivel de sofisticación, a una oferta variadísima y compleja de servicios y productos, que aterriza en su smartphone o en su tablet y le intenta seducir, mejorando por el camino la cantidad y calidad de servicios prestada por el sistema financiero [1]. La necesidad de ahorro de costes en las empresas y el surgimiento de softwares que agilicen los procesos financieros de las empresas, representan un objetivo prioritario para estas, y las proporcionan ventajas tecnológicas dentro del mercado. Las tecnologías financieras, proveen de medios flexibles y eficientes para manejar y automatizar procesos como por ejemplo pueden ser procesos para la automatización de pagos, cálculo y procesamiento de reembolsos, notificaciones para pagos atrasados o la generación de reportes personalizados con datos de interés.

Entre muchos de los procesos en la gestión de la economía de una empresa, encontramos la conciliación bancaria. La conciliación bancaria es, pues, la confrontación que se hace entre el extracto de cuenta corriente que envía el banco y el libro auxiliar de bancos de las empresas para establecer:

- Posibles diferencias ocasionadas por los registros de las operaciones.
- Corregir diferencias por errores u omisiones.
- Establecer responsabilidades.

La tarea de recopilar toda la información de las distintas bancas electrónicas que pueda tener una empresa es un procesos largo y probablemente costoso económicamente. Para realizar este proceso de manera manual, lo normal sería que varios empleados tengan que acceder una por una a cada banca electrónica, a través de las webs que tienen habilitadas para este fin, con el objetivo de extraer toda la información de movimientos y saldos que se han tenido. El ámbito de estudio de este trabajo se ha centrado la conexión entre la información bancaria y el software de gestión. Para ello, algunas bancas proporcionan mecanismos a través de API's con las que poder acceder a toda la información del usuario, así como la obtención de esta. La complejidad que produce a bancas de menor tamaño de crear una API propia para este fin, y los costes que supone, les hace inviable esta opción, por lo que el método de acceso a este tipo de bancas sería a través de la técnica de scrapping web. Este proceso permite simula el comportamiento de un usuario físico, principalmente el proceso de scrapping ingresaría con los datos de usuario a través de la web de la entidad, una vez dentro de la sesión de usuario el proceso es capaz de realizar una navegación como si se tratase de un usuario y acceder simulando pulsaciones del ratón para poder navegar a través de los menús y llegar al dato que se necesita.

Estado del arte

En 2008, se produce una crisis financiera debido al colapso de la burbuja inmobiliaria produciendo una profunda crisis de liquidez a escala mundial. Este precedente, nos sirve como punto de inflexión a lo que derivará a un mayor control de la economía de las empresas, con el fin de ahorro de costes y, con una mayor cantidad de datos e información clave, la toma de decisiones de nuestro negocio. Las finanzas tecnológicas están afectando de muchas maneras distintas en la actualidad. Por una parte ha supuesto la entrada de gran cantidad de nuevos servicios financieros distribuidos a través de diferentes canales que [2]. Entre muchos de los procesos en la gestión de la economía de una empresa, encontramos la conciliación bancaria, que es necesaria para el control total de la economía, y con el fin de evitar errores en nuestras transacciones de dinero.

La conciliación bancaria es, pues, la confrontación que se hace entre el extracto de cuenta corriente que envía el banco y el libro auxiliar de bancos de las empresas para establecer:

- Posibles diferencias ocasionadas por los registros de las operaciones.
- Corregir diferencias por errores u omisiones.
- Establecer responsabilidades.
- Mostrar el efectivo real de que dispone una empresa.

La tarea de recopilar toda la información de las distintas bancas electrónicas que pueda tener una empresa es un procesos largo y probablemente costoso económicamente. Para realizar este proceso de manera manual, lo normal sería que varios empleados tengan que acceder una por una a cada banca electrónica con el objetivo de extraer toda la información de movimientos y saldos que se han tenido. No es difícil imaginar que este problema, en empresas a partir de un tamaño medio, se vea afectadas en coste de tiempo y empleados para efectuar esta recopilación de datos. El crecimiento de una empresa va ligado con el crecimiento de sus beneficios, lo que podemos concluir de todo esto es que, teniendo las empresas como objetivo su crecimiento y obtención de mayores beneficios económicos, desencadene un crecimiento de forma paralela de la cantidad de transacciones que pasan a través de sus cuentas, posiblemente muchas de las empresas, con la facilidad de internet y la mejora de la logística mundial a la hora de gestionar productos o servicios, tengan como objetivo de expansión mercados internacionales, lo que plantea el uso de distintas bancas electrónicas para poder operar en distintos países, y no solo la expansión a nuevos mercados produce un crecimiento exponencial del número de, en este caso, bancas electrónicas que tenga que manejar una empresa, si no que la disposición de créditos y préstamos dependido de la entidad bancaria puede ser más beneficioso, desde el punto de vista financiero, en una entidad u otra.

Si multiplicamos el número de entidades bancarias, por defecto se multiplica el número de cuentas desde las que opera una misma empresa y también, multiplica el número de transacciones que necesitaría conciliar la empresa para poder tener un control total de su economía. Toda esta información puede volverse muy tediosa de procesar y la probabilidad de errores es elevada, lo que podría producir una mala gestión de la economía o incluso pérdidas de dinero. Evidentemente la contabilidad analítica es el único procedimiento o herramienta que permite recopilar toda la información de la economía de una empresa de ahí la importancia de que las empresas obtengan un software para la automatización de estos procesos específicamente diseñado para ello, automáticamente las empresas obtienen información sobre gastos e ingresos de cada uno de sus sectores de negocio.

Por tanto, no es lo mismo hacer una explotación de la información analítica con herramientas de contabilidad financiera o con herramientas diseñadas por uno mismo, que utilizar informes elaborados por casas especializadas en ello. A través de la contabilidad analítica las pymes van a disponer de una visión económica por centros de costes, frente a la contabilidad financiera que pone su foco en la empresa en general, proporcionando una visión corporativa de la situación actual de la misma. Esto nos permitirá poder tomar decisiones adecuadas, que en estos tiempos pueden suponer esa pequeña gran diferencia [3] .

Análisis de mercado

A continuación, se ha realizado un análisis de las soluciones más importantes ya existentes en el mercado:

- Tink: esta Fintech ofrece soluciones financieras de agregación de información de más de 2.500 entidades bancarias tal y como reflejan en su web oficial (<https://tink.com/products/account-aggregation/>), su principal foco de negocio se basa en la recopilación de información financiera y en el traspaso de fondos desde las distintas entidades bancarias y en pasarelas de pago, con el objetivo de realizar todo tipo de transacciones a través de un único punto.
- Kontomatik: es una solución que ofrece una recopilación de la información de distintas entidades bancarias, en el mercado español hasta con un total de 37 (<https://docs.google.com/spreadsheets/d/1jfnidiRU2s2yMfajxaFYO2GACANy5acHa11XPX2NI/edit#gid=1079391859>). Ofrece herramientas de predicción de pagos por parte de clientes y la reducción de riesgos con el análisis de comportamiento financiero (<https://www.kontomatik.com/products/financial-behaviour-analysis/>).
- Eurobits: es una empresa española que ofrece servicios de agregación bancaria, normalizando la información financiera de particulares y empresas. Ofrece una API de para la realización de pagos a través de su pasarela de pago, también ofrece servicios para transforma la factura a un modelo electrónico y la categorización de los movimientos bancarios pudiendo configurar indicadores de riesgo (KPI's) (<https://eurobits.es/>).
- Tesoralia: ofrece la agregación de la información a través de una plataforma web, entre sus servicios ofrece la generación de informes en archivos posteriormente integrables con otros sistemas empresariales, pudiendo unificar de esta manera toda la información en un único punto. (<https://www.tesoralia.com/>).
- Check it bancario: proporciona servicios de agregación financiera, a través de una plataforma online. Este software ofrece informes de conciliación bancaria, listados de clientes y proveedores y un repositorio de facturas digitales (<https://www.checkitbancario.com/>).

En el resumen anterior se han reflejado los principales focos de negocio de cada una de las soluciones que existen en el mercado actual. Basándonos a nivel de funcionalidad todos estos softwares realizan una función base de todos los agregadores financiero que es la recopilación de la información de las distintas bancas electrónicas unificándolas en una única plataforma. En los casos de las soluciones como las que nos ofrece Tink, ofrece un servicio más enfocado en las transacciones bancarias y pasarelas de pago, cumple con funcionalidades de agregador bancario, pero su foco de negocio está más enfocado hacia soluciones de pasarela de pago y centralización de transacciones bancarias. Otras soluciones como Konkomatic, también ofrecen la información de las distintas entidades electrónicas, pero en este caso, se centran en predicciones de pagos de clientes, índices de morosidad y como consecuencia reducción de riesgos a la hora de llevar a cabo cobros por parte de clientes. Por último, tenemos las soluciones como Check It, Tesoralia y Eurobits. En el caso de Eurobits ofrece todo tipo de soluciones tanto de agregación bancaria para grandes empresas, por lo que la solución para pequeñas empresas puede ser bastante costosa. Check It y Tesoralia ofrecen los mismos servicios de agregación bancario, pero con un alcance más limitado.

Marco legal

Tras el nacimiento del sector Fintech, la UE ha pretendido conceder facilidades para su expansión. Para ello se estableció una regulación financiera que afecta en mayor medida a los bancos tradicionales, especialmente a los servicios relacionados con los pagos digitales. La regulación financiera puede definirse como la introducción de una normativa que obliga a las instituciones financieras a cumplir una serie de requisitos [4] . En estos últimos años se ha ido estableciendo progresivamente ciertas regulaciones que dan mayor rigidez y estabilidad al sistema financiero. El marco legal en el que se encuentra el ámbito de estudio de este trabajo se ve afectado en una de las últimas regulaciones de la Unión Europea, la nueva directiva PSD2. Antes de comenzar a explicar la normativa PSD2, cabe mencionar a su antecesora la normativa PSD1 que, sin entrar en gran detalle, garantiza a que los usuarios que se beneficien de operaciones o transacciones monetarias reciben el importe íntegro de todas las operaciones, así mismo establece condiciones de reembolso y devolución.

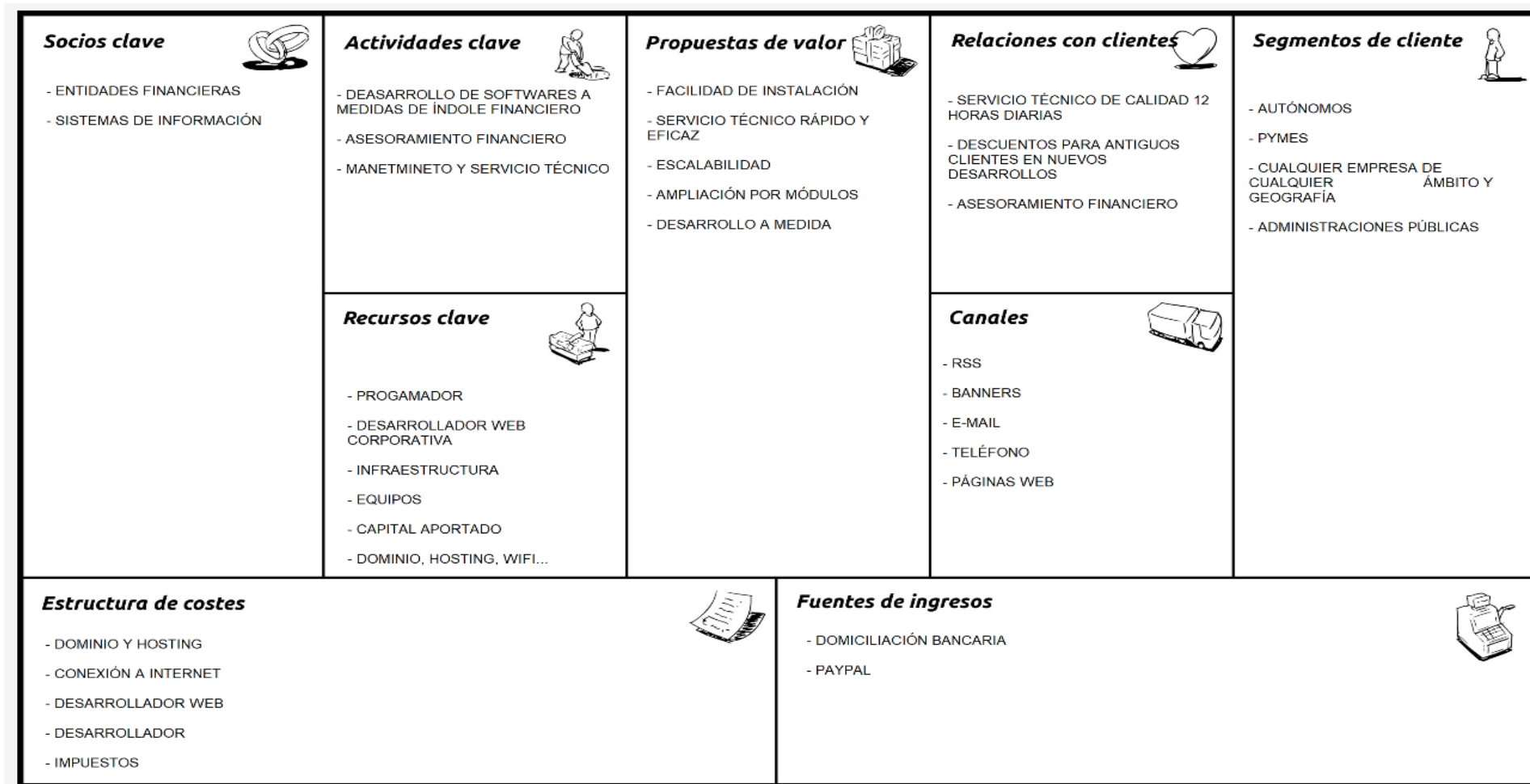
La normativa PSD2, es una directiva de Servicios de Pagos, el objetivo de este no es más que establecer dentro de la Unión Europea un mercado único de pago de esta manera fomenta la innovación, la competencia y la eficiencia. Marie Kohler (2016) definió también los objetivos que se establecieron para la PSD2:

- Formar un mercado de pagos europeo.
- Reducir las barreras de entrada para los proveedores de servicios de pago.
- Asegurar un alto nivel de protección al consumidor y de la seguridad de los pagos.
- Reducir los costes de pagos para los consumidores.
- Facilitar la creación de normas técnicas comunes, y la interoperabilidad [4] .

Dentro de los objetivos que establece la directiva, encontramos el que es más relevante para el ámbito de estudio que es la reducción de barreras de entrada para los proveedores de servicios de pago, esto quiere decir que los proveedores tendrán más facilidades de acceso a los datos bancarios del usuario, a través de APIS o mediante la autorización del propio usuario a el acceso utilizando distintos tipos de herramientas para obtener la información y realizar las operaciones oportunas.

CANVAS

Para conocer los aspectos clave del modelo de negocio utilizaremos un diagrama de Canvas, en el cual, podemos hacer visible de una forma creativa, sencilla y visual aspectos de la organización como la infraestructura, la situación financiera, oferta, clientes, de tal manera que se logren reconocer deficiencias y se analice de una forma eficiente el rendimiento del modelo. El modelo Canvas es una herramienta ideal que permite comprender de forma directa y estructura el modelo de negocio, y además puede ayudar a comprender tu propio modelo comercial y también el de tus competidores. Aunque este modelo funciona como un esqueleto del plan de negocios y establece el punto de partida para dejar que la lluvia de ideas fluya y que no sustituye a un plan de negocio detallado. Se ha decidido utilizar este modelo ya que es una metodología fácil de utilizar y en el cual de un solo vistazo se ven todos los elementos esenciales del negocio lo que sirve a su vez, como herramienta de análisis estratégico, y así poder observar debilidades y fortalezas dentro de nuestro modelo.



En el esquema anterior podemos observar el diagrama de CANVAS que representa un mapa del plan empresarial que se presenta para este proyecto, y de qué manera podría ser viable. Empezaremos hablando de la propuesta de valor, en la que cabe destacar su facilidad de instalación ya que no requiere nada más que la interfaz gráfica para poder empezar a funcionar con el sistema, teniendo en cuenta que el grosor del sistema se encuentra en la parte del servidor, el cliente solo realizaría consultas y operaciones a través de la interfaz gráfica contra la base de datos alojada en el servidor. Se ha de ofrecer un servicio técnico rápido y eficaz ya que esta tecnología está expuesta a posibles incidencias como la caída de la red, el mal funcionamiento de las bancas digitales en momentos puntuales, motivos de congestión en sus servidores cambios importantes en el diseño web que pueden alterar el comportamiento del proceso de scrapping. Ofrece además escalabilidad ya que pueden realizarse muchos módulos adicionales en función a toda la información que puede ofrecer una entidad financiera y que es de utilidad para el cliente. El sistema se basa en ampliación por módulos, por lo que no es necesario tener un proceso de scrapping completo si no que, se podrían beneficiar solo de aquellos datos que el cliente quiera conocer. Por último, se pueden realizar desarrollos puntuales a medida en función de las especificaciones de cada cliente.

El segmento de clientes es otro punto para tener en cuenta dentro del diagrama, en este caso nos enfocamos en pequeñas empresas, autónomos, pymes, administraciones públicas ya que no requiere de un desembolso alto y ayudaría automatizar ciertos procesos de su empresa, no obstante, puede ofrecerse a grandes compañías para canalizar el gran volumen de datos bancarios, las empresas pueden ser de cualquier ámbito y territorio ya que solo requiere de conexión a internet para conectar con el sistema. Los canales de difusión del sistema principalmente se centrarían en redes sociales ya que son una plataforma ideal para la difusión de material publicitario, además medios como banners en páginas webs relacionadas con el ámbito financiero o empresarial, por vía E-mail, o incluso poniéndose en contacto a través de teléfono con clientes altamente potenciales. En el punto de la relación con los clientes del sistema, se le proporcionaría un servicio técnico de calidad a lo largo del día para resolver posibles incidencias, descuentos para antiguos clientes en nuevos desarrollos de módulos de la aplicación y asesoramiento financiero personalizado.

En cuanto el modelo para la fuente de ingresos se realizaría a través de suscripciones mensuales en las cuales los clientes pueden realizar el pago a través de plataformas digitales como PayPal, o a través de la clásica domiciliación bancaria. Para que exista la posibilidad de crear un modelo de negocio, se necesitarían recursos clave para el desarrollo y mantenimiento del sistema, entre ellos el equipo de trabajo como la parte más importante por el contacto con el cliente y por la calidad a la hora de desarrollar el sistema y mantenerlo. Una infraestructura para el control del negocio, herramientas de trabajo como pueden ser equipos informáticos, material de diversa funcionalidad, un dominio web, hosting, conexión a internet son necesarias para la viabilidad del negocio. Y por último no sería nada posible si un capital inicial aportado que asuma el desembolso económico de lo mencionado anteriormente.

La actividad de la empresa se basará principalmente en el mantenimiento de las herramientas desarrolladas y dar un servicio técnico a los clientes ideal. Por otro lado, el abrir vías de desarrollo de softwares a medida en el ámbito financiero, dando asesoramiento en todas y cada una de las fases del proyecto. Para que este modelo de negocio pueda ser viable precisará de socios clave para el funcionamiento y escalabilidad de esta. El principal socio clave serán las propias entidades bancarias ya que son el centro del negocio, son las encargadas de proporcionar el dato necesario para alimentar a la aplicación. Adicionalmente tenemos a los sistemas de información como socio principal al igual que las entidades bancarias, son los encargados de integrar la información ofrecida por el sistema y así automatizar procesos financieros dentro del software de gestión del cliente. Relacionado con el punto en el que se describían los recursos clave, encontramos la estructura de costes. En este punto se destacan aquellos costes que van a ser fijos para la empresa a lo largo del tiempo y que serán necesarios para el futuro y presente de la empresa. Serán necesarios las tecnologías básicas para el funcionamiento del sistema como son un dominio y servidor, tanto como la conexión a internet y el equipo de desarrollares que darán el mantenimiento, y continuarán con el desarrollo de nuevos módulos.

Análisis de la solución

La información para las empresas es de vital importancia a la hora de tomar decisiones estratégicas que las posicionen en situaciones ventajosas de mercado con el fin de aumentar sus beneficios, por ello es importante para estas, tener mecanismo de automatización de procesos que puedan brindarle datos actualizados sobre diferentes áreas estratégicas, y que puedan ser procesados posteriormente, para la generación de informes que puedan ayudar a la toma de decisiones, así como detectar anomalías en su salud financiera. La información de carácter financiero se genera en las bancas donde las empresas operan para la realización de cobros y pagos, por lo que es necesario que los sistemas bancarios se conecten de algún modo con los softwares de gestión de las organizaciones. Este es el principal problema que se aborda, ya que son dos entes independientes tanto la banca como la empresa, por lo que no existe un modo de poder integrarlos directamente para el volcado de datos.

El volcado de datos financieros desde una banca electrónica a un software ERP se realiza a través de procesos manuales o poco semi manuales:

- **Manuales** → Un equipo de personas de la organización introduce manualmente la información de la banca al software.
- **Semi manual** → Un empleado o varios empleados descargar un fichero en un formato determinado que ofrezca la entidad financiera que posteriormente es integrado en el sistema de manera automática.

En estos dos procedimientos para realizar la integración de los datos observamos que no es un sistema autónomo, y que la actuación de la persona física es imprescindible, lo que puede llevar a un mayor número de errores y un costo en tiempo muy elevado para el empleado, si a esta casuística le sumamos el volumen de datos que puede generar una organización podemos vernos con una cantidad inasumible de información importante ya que cabe destacar que las empresas, siempre dependiendo del tamaño de estas, pueden tener múltiples cuentas en múltiples bancos, lo que multiplica exponencialmente la información. En esta solución se plantea la automatización de la descarga de información independientemente de la entidad bancaria que fuese, presentándola en un formato de fichero capaz de ser integrado por cualquier software empresarial y así, disponer de manera automática de toda esta información consolidada dentro de nuestro sistema. Con esto se pretende llegar a prácticamente cualquier tipo de empresa ya que la información financiera es común para cualquier sector empresarial por lo que el cliente objetivo podría ser cualquier organización.

Diagrama de casos de uso



Caso de uso	Iniciar sesión
Actores	Cliente
Resumen	El usuario es validado por el sistema
Precondiciones	Usuario existente y contraseña válida para el usuario
Postcondiciones	
Incluye	Validar usuario en la base de datos
Extiende	Iniciar la aplicación
Hereda de	
Flujo de eventos	
Actor	Sistema
<ol style="list-style-type: none"> 1. El cliente introduce sus datos de inicio de sesión 2. El cliente pulsa aceptar 	<ol style="list-style-type: none"> 3. El usuario y la contraseña se validan en la base de datos 4. El sistema inicia la aplicación principal

Caso de uso	Registro en la aplicación
Actores	Cliente
Resumen	Un usuario puede registrarse como cliente de la aplicación
Precondiciones	El usuario no existe
Postcondiciones	
Incluye	Validar usuario con la base de datos
Extiende	Registra usuario en el sistema
Hereda de	
Flujo de eventos	
Actor	Sistema
<ol style="list-style-type: none"> 1. El cliente rellena los datos del formulario de registro 2. Pulsa la opción registrarse 	<ol style="list-style-type: none"> 3. El sistema valida que no existe el usuario 4. El sistema registra al usuario

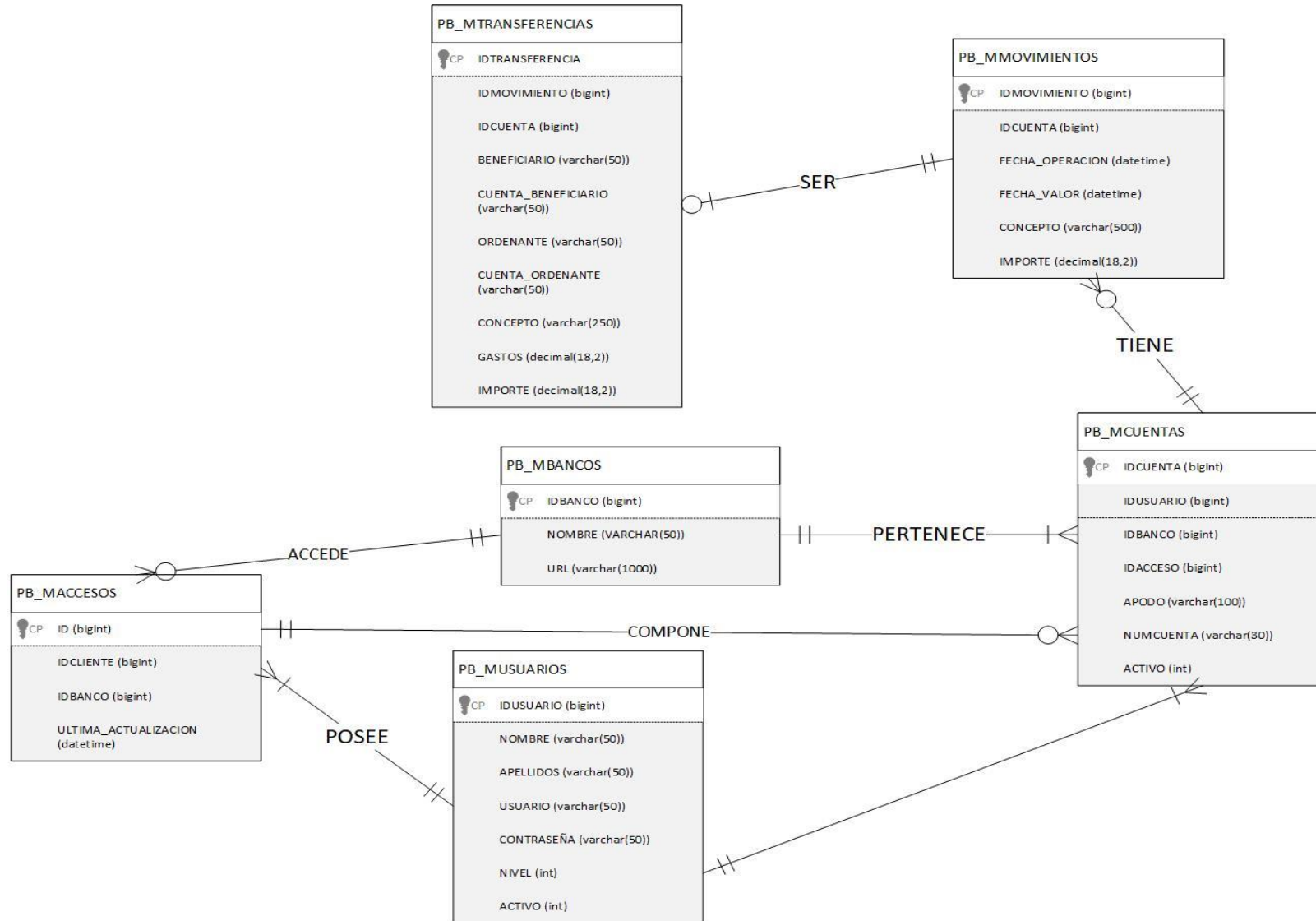
Caso de uso	Registrar Cuenta bancaria
Actores	Cliente
Resumen	El cliente da de alta una cuenta nueva
Precondiciones	La cuenta no existe en el sistema
Postcondiciones	
Incluye	Valida la cuenta bancaria en la base de datos
Extiende	Inicia la aplicación principal
Hereda de	
Flujo de eventos	
Actor	Sistema
<ol style="list-style-type: none"> 1. El cliente ingresa los datos de la cuenta bancaria 2. El cliente pulsa la opción Aceptar 	<ol style="list-style-type: none"> 3. El sistema valida la cuenta bancaria 4. El sistema registra la cuenta bancaria en el sistema

Caso de uso	Generar fichero
Actores	Cliente
Resumen	El cliente puede genera el fichero de exportación
Precondiciones	Que existan datos
Postcondiciones	
Incluye	Consultar información en base de datos
Extiende	Generar fichero
Hereda de	
Flujo de eventos	
Actor	Sistema
<ol style="list-style-type: none"> 1. El cliente selecciona en la tabla la cuenta realizando doble clic 2. Pulsa la opción generar fichero 	<ol style="list-style-type: none"> 3. El sistema consulta la información de la cuenta 4. Genera el fichero XML

Caso de uso	Refrescar
Actores	Cliente
Resumen	El cliente puede actualizar la última hora de actualización de las cuentas
Precondiciones	
Postcondiciones	
Incluye	Consulta información en base de datos
Extiende	Actualiza fecha
Hereda de	
Flujo de eventos	
Actor	Sistema
1. El usuario hace clic en la opción refrescar	2. El sistema consulta la información en base de datos 3. Actualiza la fecha de actualización

Caso de uso	Borrar campos
Actores	Cliente
Resumen	El cliente puede borrar los campos de generar fichero
Precondiciones	
Postcondiciones	
Incluye	Vaciar campos
Extiende	
Hereda de	
Flujo de eventos	
Actor	Sistema
1. El usuario pulsa la opción de Borrar campos	2. El sistema vacía los campos de generar fichero en la interfaz

Diagrama entidad-relación



El diagrama de la página anterior corresponde con el modelo de base de datos que sigue la aplicación para el almacenamiento y gestión de la información. Como podemos observar dentro del modelo de datos existen seis entidades que procederemos a explicar en los siguientes puntos.

- **PB_MUSUARIOS** → Esta entidad está compuesta por los datos básicos del usuario, registrado previamente en la aplicación, como son **NOMBRE** y **APELLIDOS**, los datos de inicio de sesión para entrar en la aplicación, **USUARIO** y **CONTRASEÑA**, y un campo **ACTIVO** con solo dos estados 1 y 0 para poder desactivar o activar la extracción de la información de la cuenta o no. Se relaciona con la entidad **PB_ACCESOS** para poder obtener la información que pertenece a un cliente, en el siguiente punto explicaremos el concepto de acceso. También observamos una relación más con la entidad **PB_CUENTAS**, de ella podemos concluir que un usuario o cliente puede tener una o más cuentas, y una cuenta puede pertenecer solo a un único cliente.
- **PB_MACCESOS** → Entendemos por “Acceso” a el nivel mínimo de información que necesita el proceso de scrapping para poder obtener la información necesaria para la extracción y consolidación de los datos capturados. Se compone por el **IDCLIENTE** de la tabla **MUSUARIOS** con el que podemos establecer la relación de que un **PB_USUARIO** pueda tener un acceso mínimo para hacer funcionar el proceso, o puede tener muchos accesos habilitados para la captura de información. Por otra parte, tenemos la relación de **PB_ACCESOS** con la entidad **PB_MNBANCOS** a través del **IDBANCO** presenten en la tabla **PB_ACCESOS** y perteneciente a la clave principal en **PB_BANCOS**, de esta forma también podremos obtener información esencial de esta entidad. Adicionalmente existe un campo **UTLIMA_ACTUALIZACION** para obtener la fecha y hora en la que el proceso de scrapping finalizó con éxito la última vez, con el fin de saber si el proceso de scrapping debe de actualizar ese acceso o por lo contrario, descartarlo. Un acceso solo puede pertenecer a un banco y solo a un único banco, por lo contrario, un mismo banco puede tener asociados ningún acceso o bien múltiples de ellos.
- **PB_BANCOS** → Como hemos comentado en la entidad **PB_ACCESOS** la conexión entre **PB_BANCOS** y **PB_ACCESOS** contempla una relación de uno a muchos, pero también contemplamos otra relación con la entidad **PB_CUENTAS**, la conexión lógica de la relación, es que a un **PB_BANCO** pueden pertenecerle muchas cuentas, y mínimo una, por el contrario, una cuenta solo puede pertenecer a una entidad bancaria.
- **PB_CUENTAS** → En esta entidad tenemos como clave principal el **IDCUENTA** para identificar cada una de estas, tiene una relación con la entidad **PB_BANCOS** como hemos explicado en el anterior punto, y otra relación con **PB_USUARIOS**. Un usuario puede abrir mínimo una cuenta y puede tener muchas, pero siempre pertenecen a un único usuario o cliente. Por último, tenemos la relación con **PB_ACCESOS**, con esta entidad obtendremos información relevante para poder saber a qué accesos de descarga pertenecen esas cuentas. Una cuenta pertenecerá a un solo acceso, con el fin de evitar las duplicidades en la descarga de datos scrapping, y un acceso puede estar compuesto por muchas cuentas o una como mínimo.
- **PB_MOVIMIENTOS** → La entidad de movimientos almacena la información básica de un movimiento bancario, su clave principal **IDMOVIMIENTO** identifica únicos registros. Los movimientos solo pueden pertenecer a una única cuenta, con las relaciones anteriores podemos deducir también que un movimiento pertenece única y exclusivamente a una cuenta, la cual solo puede pertenecer a un banco y a un cliente.

- **PB_TRANSFERENCIAS** → Esta entidad solo establece relación con el movimiento bancario, ya que a través de esta relación podemos saber exactamente a que cliente, cuenta, banco y acceso pertenecen. En esta entidad guardamos y obtenemos información relevante al detalle de la transferencia y el identificador del movimiento al que pertenece, con el objetivo de poder obtener todos los datos del modelo y poder exportar una información consolidada en el fichero de exportación XML. La relación entre la entidad **PB_TRANSFERENCIAS** y **PB_MOVIMIENTO** es una relación de uno a uno en la que una transferencia solo puede pertenecer a un único registro de movimiento bancario, y viceversa.

Diseño de la solución

En esta sección vamos a tratar de cómo puede llevarse a cabo la solución planteada, y que elementos vamos a utilizar para su realización, así como una explicación de las técnicas utilizadas. En primer lugar, vamos a hablar sobre la técnica de scrapping y para que nos va a servir dentro del proyecto.

El scrapping o también denominado raspado web, es una técnica para extraer información de sitios web simulando la navegación de una persona humana, recogiendo la información de manera automática. Dependiendo de las necesidades de la solución podemos definir diferentes vías para la obtención de estos datos:

- Uso de expresiones regulares con las que podamos detectar patrones dentro del código web.
- Peticiones HTTP al servidor remoto.
- Algoritmos de detección de patrones a través de lenguaje xpath.

En el ámbito de este proyecto se utilizará principalmente la detección de algoritmos a través del lenguaje de marcas XPATH, que es un lenguaje que permite la construcción de expresiones que recorren y procesan un tipo de documento estructurado, en formatos XML, XHTML, HTML, etc. La finalidad de este lenguaje es muy parecida al de las expresiones regulares, seleccionar puntos del documento en cuestión, con texto plano, que lleven relacionado algún atributo que los diferencie dentro de la estructura como únicos, todo este procesamiento sirve para la construcción de un fichero final.

Una vez definida la técnica que se va a utilizar pasamos a la fase de construcción de la solución y para ello utilizaremos la herramienta Visual Studio 2017. Visual Studio 2017 es un conjunto de tecnologías para el desarrollo de software basado en componentes, que permite crear aplicaciones tanto de escritorio, consola, desarrollo web, servicios Windows, etc.

Tiene integración con la plataforma de Azure para poder crear aplicaciones fácilmente, a su vez permite el desarrollo móvil entre otras opciones. La motivación para el uso de esta herramienta es que encontramos un entorno de desarrollo fácil de utilizar y en el cual ofrece ayudas para el desarrollo de código con el autocompletado de sentencias de código, así como detección eficiente de errores. Como consecuencia de la utilización de esta herramienta se decide utilizar Visual Basic NET, aunque no ofrece una especial ventaja frente a otras tecnologías como pueden ser JAVA o Python. Visual Basic .NET, es un lenguaje de programación orientado a objeto resultado de la evolución de Visual Basic. La principal característica de la evolución de las versiones anteriores de Visual Basic a la actual en .NET es el requerimiento de la programación orientada a objetos, lo que conlleva a que el programador sea más ordenado a la hora de programar, a través de reglas metodológicas de programación análogas a otros lenguajes como C++, C#, Java, etc.

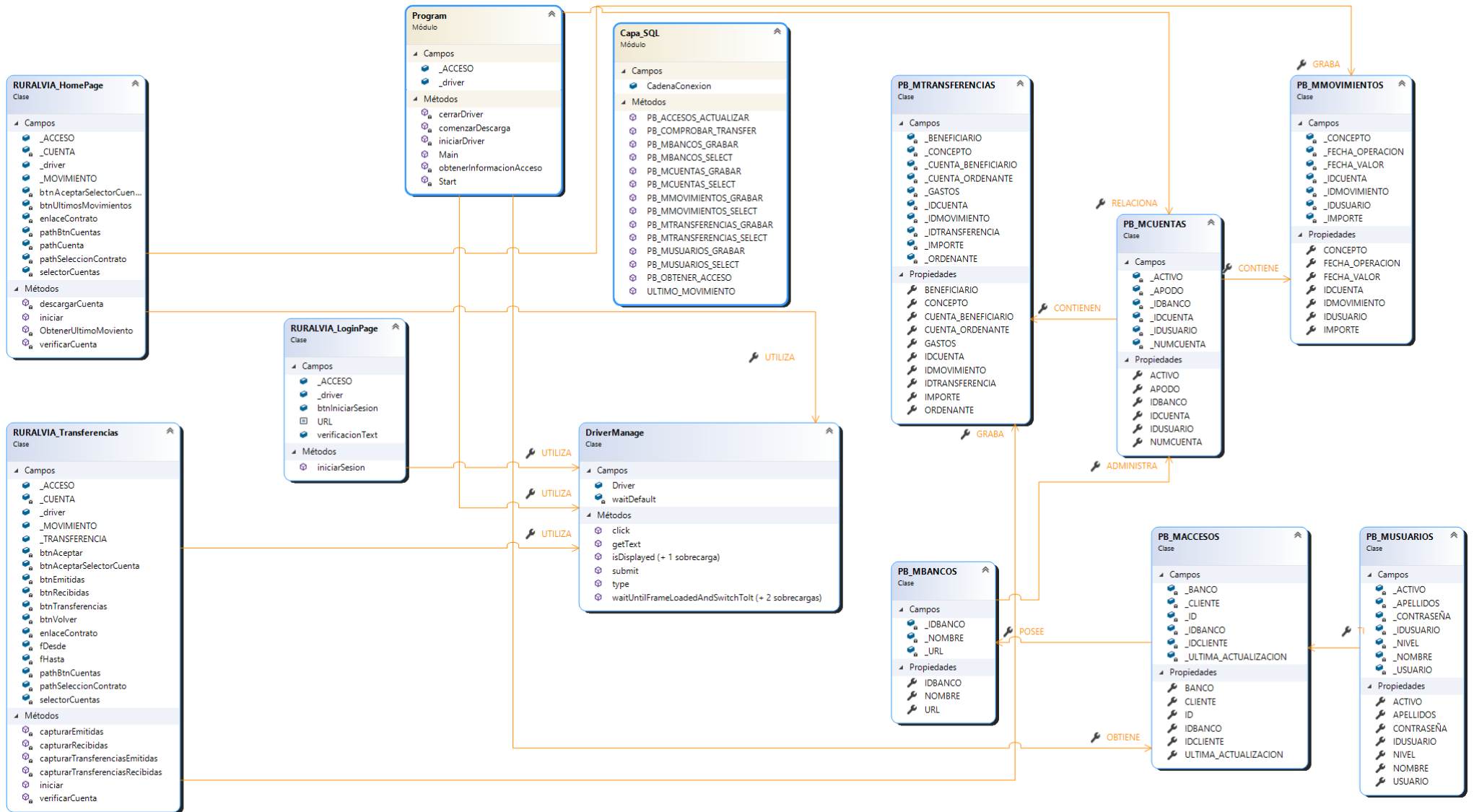
El lenguaje Visual Basic .NET brinda un completo número de características para hacer que el desarrollo de aplicaciones sea realmente rápido. El predecesor Visual Basic tenía ya una serie de características que se han mantenido en lenguajes como C# o Visual C++, como son:

1. Separación de la creación de la interfaz gráfica y el código.
2. Barra de herramientas de controles necesarios para crear interfaces gráficas mediante acciones de arrastre de componentes sobre el formulario.
3. Explorador de proyectos donde se encuentran todas las características, controles de código, propiedades, etc.
4. Permite la depuración de código durante la ejecución de esta.

Para poder implantar la metodología de scrapping, debemos de utilizar algún tipo de tecnología que nos permita controlar de forma eficiente la navegación web con la que simularemos a un usuario, en este caso utilizaremos el API de Selenium Webdriver. Selenium Webdriver es un entorno de pruebas para aplicaciones basadas en web. Selenium proporciona una amplia librería o API, con métodos y funciones para realizar todo tipo de pruebas web basadas en múltiples lenguajes de programación como puede ser Java, Python, PHP, Perl, Ruby, Groovy y en nuestro escenario VB .NET. También es compatible en sistemas operativos Windows, Linux y OSX, y se puede ejecutar en prácticamente cualquier navegador incluidos los más utilizados como Safari, Internet Explorer, Chrome y Opera. Selenium también permite las peticiones POST y GET a través del navegador web, y obtener recursos de esta. [5]



Diagrama de clase



En la imagen de la página anterior podemos observar el diagrama de clases de la aplicación de scrapping, encargada de la extracción de datos y procesamiento de estos. A continuación, se procederá a la explicación de cada una de sus partes.

- Estableceremos el módulo `Program` como el inicio del proceso. Para poder iniciarlo, primero debemos obtener la información a través del módulo `Capa_SQL`, este módulo es el encargado de conectar y obtener la información esencial de la base de datos. En la `Capa_SQL` podemos inicializar todos los objetos de cada una de las clases, con la información que necesitaremos extraer, mediante el paso de parámetros opcionales en función de las necesidades en el momento de la consulta en base de datos, así como también es el encargado de realizar los cambios en registros e inserciones de nuevos elementos. El atributo necesario para el funcionamiento de la `Capa_SQL` es la cadena de conexión a la base de datos, la cual, está introducida en el archivo de configuración `app.config`. La nomenclatura en el archivo de configuración en formato XML para la cadena de conexión sigue la siguiente tipología.

```
<connectionStrings>
<add name="sqlDatos" connectionString="server=" NOMBRE DEL SERVIDOR"; database="
NOMBRE DE LA BASE DE DATOS"; Integrated Security=SSPI; user
;id="USUARIO";password="CONTRASEÑA" " />
</connectionStrings>
```

- El módulo `Program`, utiliza como elementos, el objeto de la clase `PB_ACCESOS` y el objeto `_driver`. El objeto `driver` es una instancia de la dll de `Selenium Web Driver` tecnología de la cuál hemos hablado en puntos anteriores. Esta dll, permite utilizar un objeto como controlador del navegador web que utilizaremos para poder realizar el proceso, también es el encargado de extraer en formato texto toda la información que busca el proceso de scrapping. Esta extracción de información se basa a través de rutas utilizando lenguaje XPATH. Xpath es un lenguaje que permite construir expresiones que recorren y procesan un tipo de documento con un formato estructurado como XML o en este caso el lenguaje de etiquetas, por las que las webs están compuestas, HTML. El objeto `Selenium Web Driver`, ya contiene las funcionalidades necesarias para el manejo de eventos dentro del navegador, pero con el objetivo de añadir un nivel de abstracción para poder trabajar más fluidamente se decide crear la clase `DriveManage`, que contiene funciones útiles que se utilizan a lo largo del código para un manejo más sencillo de este, el objeto `driver` es una instancia de esta última clase, esta, solo necesitará la instancia inicial de `Selenium Web Driver` y un tiempo en milisegundos dependiendo de la necesidad de espera que necesitaremos para controlar los tiempos de carga de las páginas webs. En el inicio de `Program`, obtiene la información del acceso utilizando la función correspondiente dentro del módulo de `Capa_SQL`, posteriormente se utilizan las funciones de iniciar driver para crear la instancia del controlador web, y se continua con la función `comenzarDescarga`, con la que se decidirá en función de la información del acceso, que banco y por lo tanto que clase vamos a instanciar para iniciar la descarga. En este caso de estudio se ha realizado la codificación de extracción de información con la página de demostración del banco RURALVIA, pero esta técnica es equivalente para todas las demás bancas ajustándola a las peculiaridades de cada una.

- Una vez que `Program` obtiene toda la información y comienza la descarga, se ejecutará la clase `RURALVIA_LoginPage`.
- `RURALVIA_LoginPage`, es la clase encargada de iniciar una instancia del navegador web para después ingresar la URL de la página de la entidad bancaria. Una vez aparece el formulario de verificación web, ingresará usuario y contraseña y presionar el botón de inicio de sesión, de esta manera accedemos dentro de la sesión del usuario. La clase verificará a través del campo `verificationText`, si el inicio de sesión es el correcto, este proceso se realiza buscando dentro de la página si encontramos el mensaje de “*La contraseña o usuario no ha sido introducido correctamente*”, si pasado un tiempo de espera que realiza el proceso mediante “`Thread.Sleep(milisegundos)`”, no aparece este mensaje, la clase verifica a `Program`, que se ha iniciado sesión con éxito, por lo tanto, puede continuar la descarga.
- Una vez verificada la sesión del usuario, procedemos a llegar hasta la información, para posteriormente extraerla y almacenarla. La clase `RURALVIA_HomePage`, es la encargada de esta extracción de información, cuenta como campos principales el objeto acceso, que obtiene la información esencial con la cual podemos saber al cliente que pertenece el acceso, banco, y la última actualización de este. Adicionalmente tenemos dos objetos privados propios de la clase que son `MOVIMIENTO` instancia de la clase `PB_MOVIMIENTOS` y `CUENTA`, instancia de la clase `PB_CUENTAS`. Lo primero que va a realizar la clase es seguir el camino lógico que seguiría un usuario para mostrar el desplegable de cuentas, el controlador buscar a través del XPATH la etiqueta que contenga la palabra clave “ver cuentas” una vez el driver haya localizado está etiqueta ejecutará la función `click` que desencadena el evento en cuestión simulando un `click` con el ratón, en caso de que no haya podido encontrar la etiqueta con el texto clave, el driver devolverá por defecto un error de `timeout` o tiempo de espera que es recogido por la aplicación saliendo del proceso. Una vez cargada la pantalla de ver cuentas, la web bancaria mostrará un desplegable con cada una de las cuentas disponibles, este desplegable o selector se seleccionará de la misma forma que se ha detectado el texto de ver cuentas, la mecánica para localizar textos siempre es la misma siguiendo una ruta XPATH. Aquí podemos ver el ejemplo de cómo buscamos un texto.

```
//a[contains(text(),'CUENTAS')]
```

Las dos contra barras indican que tienen que buscar dentro del `<BODY>` de la página, una vez estamos situados dentro de este, queremos que busque una etiqueta `<a\>` cuyo texto contenga la palabra cuentas, esta nomenclatura en concreto situará el foco en el enlace que cumpla estas condiciones que han de ser los más concretas posibles para evitar seleccionar otros elementos. Una vez accedemos a una de las cuentas, podemos observar en la siguiente pantalla la tabla con los movimientos, por lo tanto, la información que debemos de capturar. En este caso de estudio, al usar una banca con una versión de demostración, no sería necesario filtrar por fechas ya que aparece toda la información en pantalla, pero en casos reales y utilizando siempre el mismo método, obtendríamos la ruta de los componentes webs de fechas, para así poder modificarlos y filtrar entre las fechas que se precisa con el fin de reducir el nivel de información no deseada en pantalla y evitar errores, pero en ambos casos, debemos de comprobar la fecha de los movimientos, y la fecha de última actualización del acceso, verificando de esta manera que la información mostrada en pantalla es la más actual siempre que la que tenemos guardada en base de datos.

Una vez realizado los pasos anteriores recorreremos cada una de las filas de la tabla, de esta manera podremos tener cada registro de esta. Para poder recorrer cada uno de los movimientos, se utiliza una estructura `for`, en el cual, por cada una de sus vueltas hay que verificar diferentes parámetros:

1. Primero definimos una ruta lo más concreta posible, pero que a su vez sea capaz de identificar un listado de etiquetas en concreto, en este caso nos interesan las etiquetas `<table>`, que identifica el contenedor general, `<tr>` que hace referencia a cada una de las filas de la tabla y dentro de esta la etiqueta `<td>` que corresponden a cada una de las columnas. Basándonos en esto podemos construir una ruta como esta `\\table\\td[index]`, en esta ruta estaremos accediendo a la tabla de movimientos, posteriormente accedemos a cada una de sus filas utilizando el variable `index` de la estructura `for`.
2. El siguiente paso es verificar que el siguiente movimiento de cada fila seleccionada utilizando la ruta anterior, existe en pantalla, en caso contrario habremos llegado al final de esta y terminamos de obtener la información.
3. En la casuística de que exista una fila con movimientos la mecánica para acceder a cada uno de sus campos es de la misma forma que en el paso 1, accediendo a través de la ruta de ejemplo siguiente:

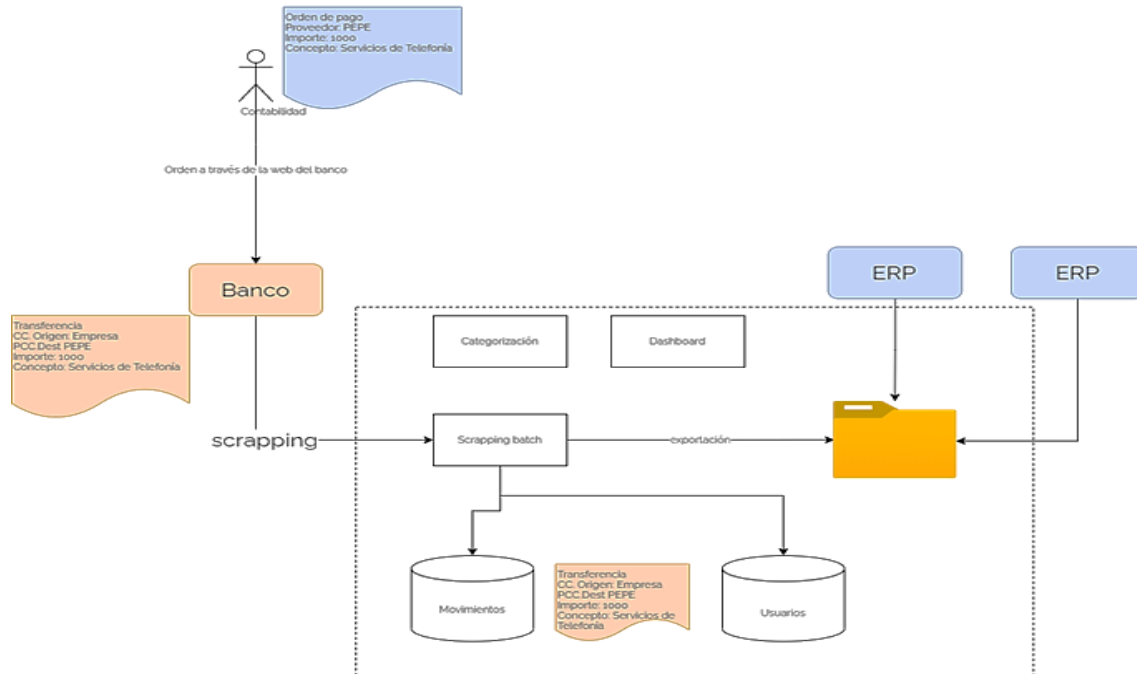
```
\\table\\td[index]\\td[posicion_campo]
```

La posición de cada campo en la tabla es una posición fija por lo que podremos acceder a la ruta de cada uno de los campos y extraer el texto que contiene cada etiqueta, de esta manera sabemos la información del campo, y cual corresponde, construyendo de esta manera el objeto movimiento.

4. Una vez construido el objeto, se procede a su inserción en base de datos a través del módulo `CAPA_SQL`.
 - Cuando el proceso de extracción de los datos y su posterior inserción en base de datos concluye, se devuelve el control a `Program`, si la descarga ha salido correctamente, el proceso continúa con la instancia de la clase `RURALVIA_Transferencias`, que es la encargada de realizar exactamente el mismo proceso anterior, pero esta vez seguirá la ruta lógica para ver la información de transferencias utilizando las mismas funcionalidades que la anterior extracción de información.
 - Una vez termina el proceso de extracción de transferencias, se devuelve el control del proceso a `Program`, este será el encargado de realizar la convergencia de datos entre movimientos y trasferencias
 - La relación de los datos, en este caso, de transferencias bancarias al detalle, con su respectivo movimiento bancario, que ofrece solo información básica, se realiza a continuación de que tengamos la información de ambos ámbitos consolidados en base de datos, con lo que obtendremos todas aquellas transferencias que no estén asociadas a ningún `ID_MOVIMIENTO`, lo que quiere decir que esas transferencias no han sido relacionadas con su respectivo movimiento. En la descripción de estos, está detalla la referencia de la transferencia, la cual utilizamos para buscar dentro del concepto de estas, en caso de coincidencia podemos deducir que ese movimiento bancario corresponde con el detalle de la transferencia. Una vez se realiza este proceso con todos los registros, se da por concluido el proceso de extracción de la información para posteriormente exportara al formato estructurado XML y poder ser legible para distintos ERP's

Esquema de funcionamiento

Sistema de agregación de información financiera para entornos empresariales



Descripción técnica

En esta sección vamos a entrar más en profundidad en la dinámica del código. El proceso se ejecuta automáticamente a través de una tarea programada, llamando al ejecutable de la aplicación. La aplicación es un programa de consola, por lo que acepta parámetros, que son los necesarios para iniciar un proceso de scrapping. En este caso pasaremos como argumentos `-acceso 1`, esto quiere decir que ejecutaremos el acceso con el número identificador 1, que corresponderá a un usuario en concreto y a una entidad bancaria única. De esta forma la clase principal `main` de la aplicación puede obtener la información necesaria. El siguiente paso a realizar, es la obtención de la información del acceso, se realiza a través de la ejecución del procedimiento almacenado `WF_PB_MACCESOS_SELECT`. Este procedimiento obtiene como parámetro de entrada la identificación del acceso y retorna información tal como el `IDCLIENTE`, `IDBANCO`, `CLIENTE`, `BANCO` y `ULTIMA_ACUTALIZACION`, con esta información puede comenzar el proceso de descarga. Al comenzar la descarga nos encontramos una estructura `SELECT CASE`, este tipo de estructura permite seleccionar distintos caminos de código por el que continuará nuestro proceso, en este caso seleccionamos el caso que nos haya proporcionado el campo `BANCO`, como por ejemplo `BANKIA`. Al seleccionar una entidad bancaria en concreto, ejecuta las clases específicas para la extracción de información de esa banca. Antes de comenzar el proceso se inicializa el controlador de `Selenium web Driver`.

```
driver = New ChromeDriver(System.AppDomain.CurrentDomain.BaseDirectory)
driver.Manage.Window.Maximize()
```

En las líneas anteriores de código podemos observar cómo inicializamos el objeto `_driver` realizando la instancia del controlado para el navegador Google Chrome, pasando como parámetro la dirección donde se encuentra almacenado el ejecutable que permite el control del navegador, en nuestro caso estará dentro del propio directorio de la aplicación. Una vez inicializado el objeto, toma el control de la instancia del navegador ya creada, la muestra en pantalla, y posteriormente maximizamos a pantalla completa la ventana, para evitar posibles cortes de texto en el caso de que no se adapte la página web a la ventana del navegador.

Una vez tenemos el control sobre el navegador se realiza la instancia de la clase `Login` de la entidad correspondiente para verificar el inicio de sesión.

```
If loginPage.iniciarSesion() Then
```

De esta forma ejecutamos, en la clase `login` y a través de su objeto instanciado `loginPage`, la función `iniciarSesión`. Esta clase recibe como parámetros el controlador y el objeto `ACCESO` que hemos obtenido anteriormente en el inicio del programa. El controlador navega hacia la URL de la entidad bancaria.

```
_driver.Navigate.GoToUrl(URL)
```

En ciertas ocasiones, podemos encontrar estructuras en el código HTML como es la etiqueta `<FRAME>`. Estas etiquetas crean un contenedor almacenando dentro de la estructura del código HTML, esto presenta un obstáculo, ya que si queremos obtener la información de dentro de esta etiqueta previamente debemos realizar una carga como contenedor principal que anteriormente era la etiqueta `<BODY>`.

```
_driver.WaitUntilFrameLoadedAndSwitchToIt("atributo name del frame")
```

Esta línea de código realiza la función de cargar el `frame` seleccionado como contenedor principal y de esta manera poder acceder a toda la información de su interior. Una vez el controlador haciendo uso de la clase `DriveManage` introduce las credenciales, realiza un clic sobre el botón iniciar sesión, el proceso duerme un segundo haciendo uso de `Thread.Sleep(milisegundos)` con el fin de dar tiempo a la página web en verificar la información, una vez verificamos que no parece ningún mensaje en referencia a credenciales mal introducidas, retornamos un tipo de dato boolean con valor `true`, verificando así el inicio de sesión. Una vez realizada la verificación se instancia la clase que inicia la función `iniciar` y de este modo comienza la navegación a través de las opciones de menú de la página web basándose en rutas XPATH.

```
Dim selectorCuentas As By = By.XPath("//select")
```

En la línea anterior observamos como inicializamos una variable `selectorCuentas` como un tipo XPATH, en el que introducimos en su interior la ruta del desplegable con las cuentas para poder desplegarlo.

```
Dim recorrerCuentas As By = By.XPath("//select//option[" + contador.ToString + "]")
```

La variable `recorrerCuentas` contiene la ruta que se ha de recorrer utilizando un bucle `for` para cada una de las opciones del selector. Una vez seleccionada la primera opción del desplegable, obtenemos el texto de cada cuenta, la cuenta se verifica en base de datos para tener constancia de que existe y que debemos de realizar el proceso de extracción de esta. En el proceso de descarga o extracción de la información de la cuenta tendremos que detectar la tabla dentro del código HTML donde se encuentran los datos. Nos encontramos en este punto con la problemática de no saber cuándo terminamos de capturar movimiento por lo que declaramos una variable de tipo Boolean, `finMovimientos` admitiendo solo valores `true` o `false`, con la finalidad de saber en qué momento debemos cambiar de estado, teniendo en cuenta que la variable se inicializa a `false` y de que utilizaremos un bucle `Do While`, en el que solo cuando cambie de estado la variable `finMovimientos` detendremos el proceso de descarga y habremos llegado al final de los movimientos. Una vez entramos dentro del bucle y haciendo uso de una variable contador de tipo `integer`, que incrementará en una unidad por cada vuelta del bucle, podemos ir recorriendo cada una de las filas como observamos en la siguiente línea de código.

```
Dim pathFila As By = By.XPath("//div[contains(@id,'BODY_LISTA')]/table//tbody//tr[" & filas & "]")
```

Antes de empezar a capturar la información dentro de un objeto de tipo `PB_MOVIMIENTO`, debemos confirmar a través de la funcionalidad `isDisplayed("ruta XPATH")`, si la ruta que buscamos dentro del código esta visible o disponible, en el caso de que no encontremos la ruta, verificamos de que no existen movimiento dentro de los filtros establecidos de fecha, por el contrario habremos encontrado información que debemos de capturar y una vez almacenados los campos dentro del objeto `PB_MOVIMIENTO` realizamos una sentencia `INSERT` a través del procedimiento almacenado de `MMOVIMIENTOS_INSERT` con la que almacenaremos los datos. Una vez llegamos al final de los movimientos y verificando de que no hemos encontrado una nueva fila y cambiando el estado de la variable `finMovimientos`, finalizamos el bucle. Este proceso se realiza por cada una de las cuentas del selector que hayan sido verificadas dentro del sistema con éxito. El siguiente paso se realiza a través de la clase específica para capturar transferencias bancarias, la mecánica es la misma que la vista anteriormente, pero con la peculiaridad de las rutas XPATH específicas para estas.

En el ejemplo de código se muestra el bucle que permite la captura de la información y su grabación:

```

Do While finMovimientos = False

    Dim pathFila As By = By.XPath("//div[contains(@id,'BODY_LISTA')]/table//tbody//tr[" & filas & "]")

    If dr.isDisplayed(pathFila) Then

        _MOVIMIENTO.FECHA_OPERACION =
        CDate(dr.getText(By.XPath("//div[contains(@id,'BODY_LISTA')]/table//tbody//tr[" & filas &
        "]/td[1]")).Replace(" ", "").Trim)

        If Not _ULTIMO_MOVIMIENTO.FECHA_OPERACION.Year.ToString.Equals("1") AndAlso
        (_MOVIMIENTO.FECHA_OPERACION >= _ULTIMO_MOVIMIENTO.FECHA_OPERACION) Then

            finMovimientos = True
            Exit Do

        Else

            _MOVIMIENTO.FECHA_VALOR =
            CDate(dr.getText(By.XPath("//div[contains(@id,'BODY_LISTA')]/table//tbody//tr[" & filas &
            "]/td[2]")).Replace(" ", "").Trim)

            _MOVIMIENTO.IMPORTE =
            CDec(dr.getText(By.XPath("//div[contains(@id,'BODY_LISTA')]/table//tbody//tr[" & filas &
            "]/td[3]")).Replace(" ", "").Trim)

            _MOVIMIENTO.CONCEPTO = dr.getText(By.XPath("//div[contains(@id,'BODY_LISTA')]/table//tbody//tr["
            & filas & "]/td[5]")).Replace(" ", "").Trim

            _MOVIMIENTO.IDCUENTA = CInt(_CUENTA.IDCUENTA)
            _MOVIMIENTO.IDUSUARIO = CInt(_ACCESO.IDCLIENTE)
            _MOVIMIENTO.IDMOVIMIENTO = 0

        End If

    Else

        finMovimientos = True
        Exit Do

    End If

    Capa_SQL.PB_MMOVIMIENTOS_GRABAR(_MOVIMIENTO)
    _MOVIMIENTO = New PB_MMOVIMIENTOS
    filas = filas + 1

```

La siguiente fase la clase `main` del programa, es proceder a la categorización de la información. La categorización funciona buscando la referencia del concepto de la transferencia en el campo concepto del movimiento a través de una sentencia `SELECT`, con esto obtenemos un registro único con lo que podemos asumir que el detalle de esa transferencia corresponde con el movimiento, e insertamos el campo `idmovimiento` en el campo correspondiente `PB_MTRANSFERENCIAS`. Por último, se cierra el controlador y se actualiza la hora de finalización del proceso con éxito en el campo `UTLIMA_ACTUALIZACION` de la tabla `PB_ACCESOS` dándolos por concluido.

```
_driver.Close()  
_driver.Dispose()  
_driver = Nothing
```

Ejemplo: Documento XML:

```
<libro>
  <titulo>Dos por tres calles</titulo>
  <autor>Josefa Santos</autor>
  <capítulo num="1">
    La primera calle
    <parrafo>
      Era una sombría noche del mes de agosto...
    </parrafo>
    <parrafo destacar="si">
      Ella, inocente cual
        <enlace href="enlace">mariposa</enlace>
      que surca el cielo en busca de libaciones...
    </parrafo>
  </capítulo>
  <capítulo num="2" public="si">
    La segunda calle
    <parrafo>
      Era una obscura noche del mes de septiembre...
    </parrafo>
    <parrafo>
      Ella, inocente cual
        <enlace href="enlace">abejilla</enlace>
      que surca el viento en busca del néctar de las flores...
    </parrafo>
  </capítulo>
  <apéndice num="a" public="si">
    La tercera calle
    <parrafo>
      Era una densa noche del mes de diciembre...
    </parrafo>
    <parrafo>
      Ella, cándida cual
        <enlace href="enlace">abejilla</enlace>
      que surca el espacio en busca de bichejos para comer...
    </parrafo>
  </apéndice>
</libro>
```


Estructura Arborea:

```

/
+---libro
|
+---título
|   |
|   +---(texto)Dos por tres calles
|
+---autor
|   |
|   +---(texto)Josefa Santos
|
+---capítulo [num=1]
|   |
|   +---(texto)La primera calle
|   |
|   +---parrafo
|   |   |
|   |   +---(texto)Era una sombría noche...
|   |
|   +---parrafo [destacar=si]
|   |   |
|   |   +---(texto)Ella, cual inocente
|   |   |
|   |   +---enlace [href=enlace]
|   |   |   |
|   |   |   +---(texto)mariposa
|   |   |
|   |   +---(texto)que surca el cielo en busca de libaciones...
|
+---capítulo [num=2, public=si]
|
+---(texto)La segunda calle
|
+---parrafo
|   |
|   +---(texto)Era una obscura noche...
|
+---parrafo
|
+---(texto)Ella, cual inocente abeja...

```


Arquitectura Cliente-Servidor

En el diseño de este software se ha optado por una arquitectura cliente-servidor básica, ya que no son necesarias capas intermedias o externas. El modelo cliente-servidor reparte los recursos entre el proveedor o proveedores, que en este caso sería el servidor, y los demandantes o clientes que acceden a estos recursos. Los beneficios que aporta este tipo de arquitectura son la centralización tanto en los accesos, como en recursos y la integridad de los datos que son gestionados directamente por el servidor. Permite escalabilidad y puede ser escalable en cualquier momento, es de fácil mantenimiento, la existencia de tecnologías suficientemente desarrolladas y diseñadas para este tipo de arquitectura y dificulta el rastreo. Puede tener problemas de congestión y tanto el software como el hardware son muy determinantes [7].

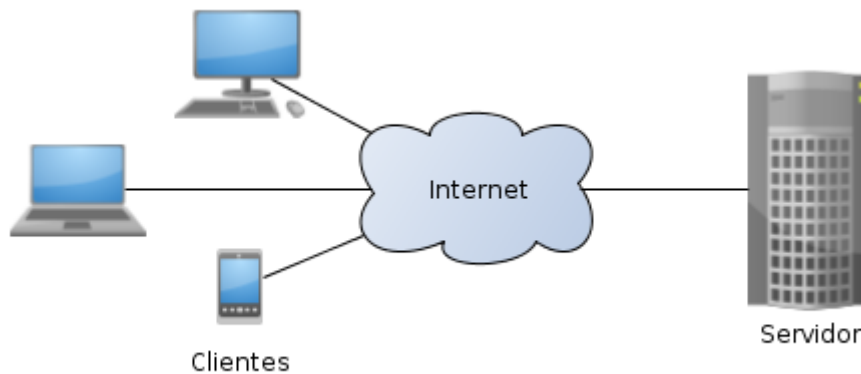
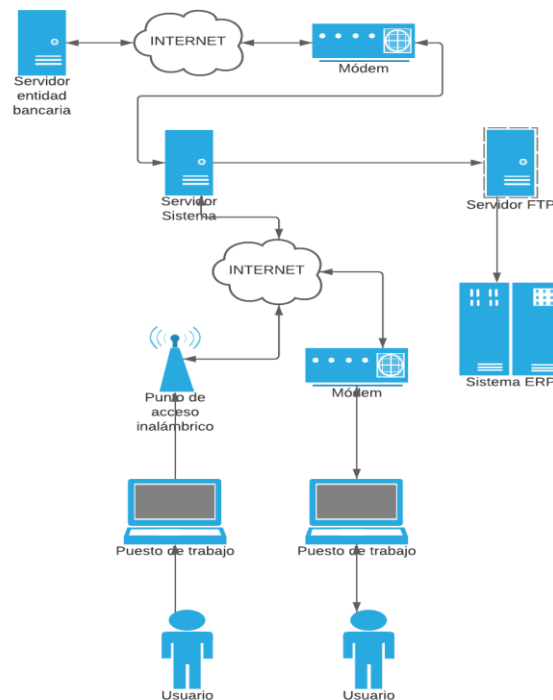


Diagrama de red

En esta sección se van a definir los componentes que conforman la arquitectura de red de la aplicación. En primer lugar, utilizaremos una máquina que actuará como servidor del sistema, en este servidor encontramos las configuraciones de las tareas programadas que se utilizarán para la automatización del proceso de scrapping, así como la base de datos donde almacenaremos toda la información. El servidor de la aplicación como acabamos de comentar es el encargado de ejecutar el proceso de scrapping por lo que estará conectado a un modem que realizará la conexión entre el servidor e internet, ya que es necesario para poder acceder a las distintas bancas electrónicas y extraer la información de estas.

En segundo lugar, tendremos dos tipos de conexiones que utilizarán las aplicaciones de los usuarios finales, para obtener la información desde el servidor, los usuarios pueden conectarse o bien de forma inalámbrica, o bien pueden conectarse a través de una red cableada con acceso internet, la aplicación se conecta a través de estas conexiones para obtener la información. Por último, el servidor del sistema se comunica a un servidor FTP, en este caso de estudio en particular, no sería necesaria para este paso ningún tipo de conexión, ya que el fichero de exportación que genera el proceso de scrapping se almacena de forma local, pero puede adaptarse esta parte a las necesidades del cliente final, bien dejando el fichero de exportación en un servidor remoto, carpeta compartida, adjunto en un email, etc. El sistema final del cliente, el cual es el encargado de integrar los ficheros de exportación, se conectará de la manera acordada, al lugar de almacenamiento de este fichero final, en este caso a un servidor FTP, dependiendo de donde se encuentre este servidor, si es una red local o externa, se necesitará conexión a internet o no.



Funcionamiento del sistema de scrapping

En el sistema planteado debemos tener en cuenta todos los elementos externos que puedan afectar a su funcionamiento o integridad de los datos. Como elemento externo tendremos las entidades bancarias, que son las encargadas de centralizar la información de las operaciones que realiza el usuario tales como transferencias, recibos, remesas, pagos, etc. En el sistema externo interactúa el usuario para realizar las operaciones y tener un punto de acceso a esta información. En el siguiente paso del funcionamiento del sistema, entraremos en la parte interna de este, en la que entra en funcionamiento el proceso de extracción. Para realizar con éxito el proceso de extracción (“scrapping”) se ha categorizado a los usuarios por medio de accesos, estos accesos son número identificadores que dentro del modelo de datos del sistema podemos obtener usuario, contraseña y URL de acceso a los datos, así como el nombre de la entidad bancaria a la que vamos a acceder.

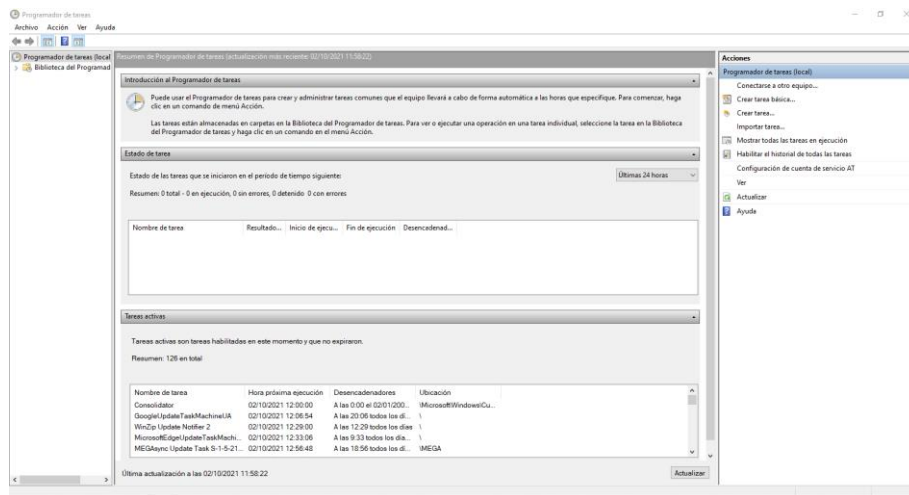
El proceso de scrapping estará programado para que se ejecute a partir de las 00:00 de cada día con el fin de asegurar que la información del día transcurrido ya está consolidada y no va a efectuarse una operación posterior. Una vez el proceso haya extraído toda la información, se inserta en una base de datos para poder procesarla posteriormente. Como tercer elemento importante dentro del sistema, tenemos un proceso de categorización de la información. El proceso se encarga de relacionar todas las operaciones del usuario de transferencias, recibos, remesas, etc., con su correspondiente operación. Para hacerlo el proceso busca las palabras claves como referencias, importes, ordenantes o números de cuenta, de esta forma podemos obtener una relación de uno a uno con sus detalles, y así de esta forma tener toda la información posible por cada operación de una forma automática y eficiente. Más adelante, se genera un fichero en formato XML con la información de cada una de las operaciones detalladamente y este es depositado en una carpeta compartida o servidor FTP, o cualquier ubicación con la finalidad de que un sistema financiero empresarial acceda a ese fichero e integre toda la información detallada dentro del sistema de contabilidad de la empresa.

Implantación

Uno de los puntos fuertes del proyecto es la facilidad de su despliegue o implantación en el sistema, ya que no requiere de conocimientos avanzados y se puede utilizar las funcionalidades que proporciona cualquier versión de Windows con interfaces fáciles de utilizar e intuitivas. Para la implantación del sistema vamos a utilizar un servidor Windows Server ya que nos proporciona el entorno compatible para nuestra base de datos SQL Server 2012. La elección de esta base de datos se debe a sus siguientes ventajas:

1. Escalabilidad, estabilidad y seguridad.
2. Soporte de procedimientos almacenados.
3. Entorno gráfico para su administración.

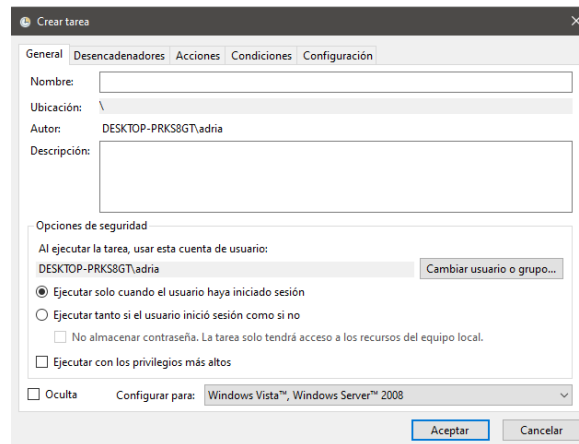
El sistema operativo Windows, y en todas sus versiones ofrece la posibilidad de crear tareas programadas, las cuales utilizaremos para el automatismo del proceso de scrapping.



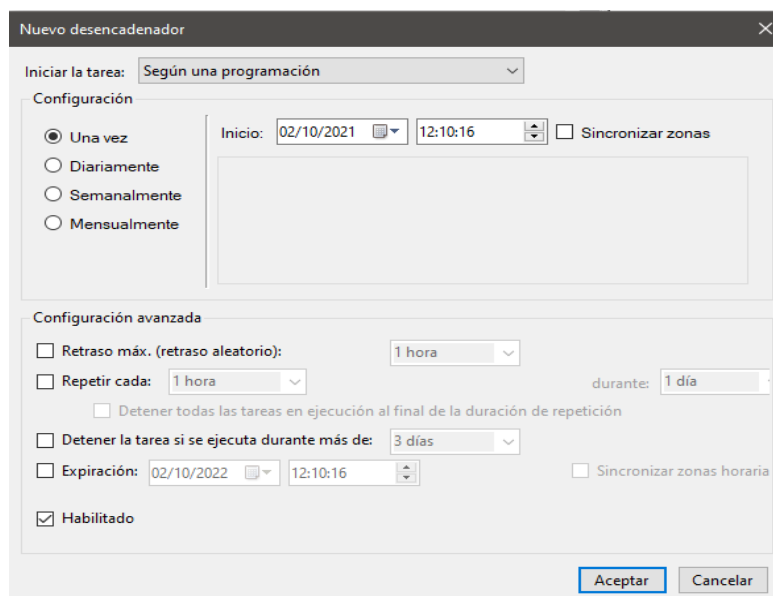
Interfaz del programador de tareas 1

El proceso de scrapping, al ser un programa de consola, es capaz de recibir los parámetros a través de comandos, los cuales se permiten ejecutar dentro de las tareas programadas. En las opciones de crear una tarea encontramos distintos campos y opciones que debemos configurar para la automatización del sistema.

1. El primer paso es la configuración general, donde realizaremos una descripción básica del proceso y desde que usuario de Windows queremos ejecutar, de esta forma podemos configurar varios usuarios a las que denominaremos sesiones, en las que poder configurar un conjunto de accesos del sistema, los cuales se ejecutarán uno tras otro en el proceso de scrapping.

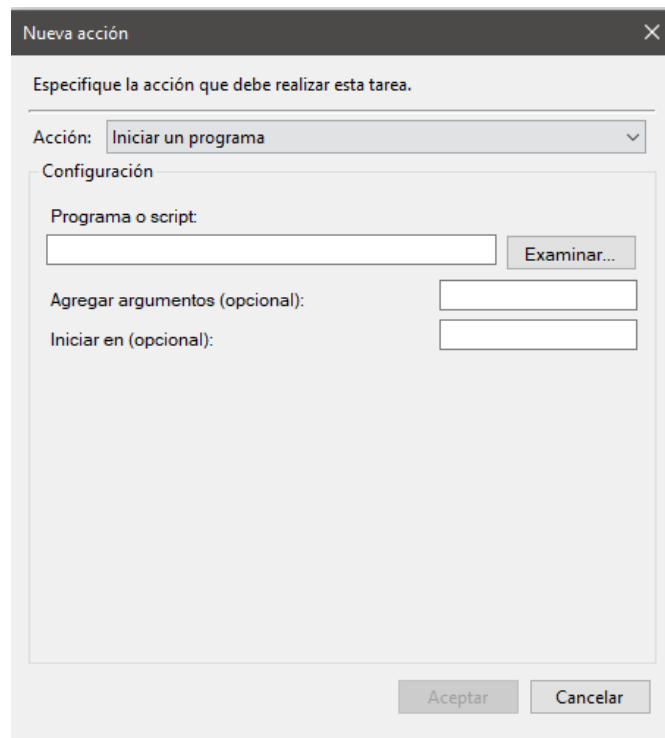


2. El siguiente punto que debemos configurar son los desencadenadores, en esta parte de la configuración nos limitaremos a programar la frecuencia en días y la hora a la que se ejecutará esta tarea en el servidor. También podemos configurar cuando debe finalizar la tarea el sistema operativo con el objetivo de no dejar el proceso atascado en caso de error



3. En la parte de acciones seleccionaremos la opción “Iniciar un programa, esta será la orden que realizará la tarea. También hay que indicarle sobre qué programa se ha de realizar utilizando la ruta de nuestro sistema al ejecutable de la aplicación. Por último, configuraremos los argumentos de la aplicación, puede ser de dos formas:
 - acceso 1* → Ejecuta solo un acceso del proceso de scrapping
 - sesion 1* → Ejecuta todos los accesos pertenecientes a la sesión 1

Con esta configuración podemos completar la automatización con todas las tareas adicionales que requiera nuestro sistema. La interfaz de usuario puede instalarse directamente sobre el servidor, configurando un escritorio remoto para cada usuario y poder conectarse de forma individual al servicio



Nueva acción

Especifique la acción que debe realizar esta tarea.

Acción: Iniciar un programa

Configuración

Programa o script: Examinar...

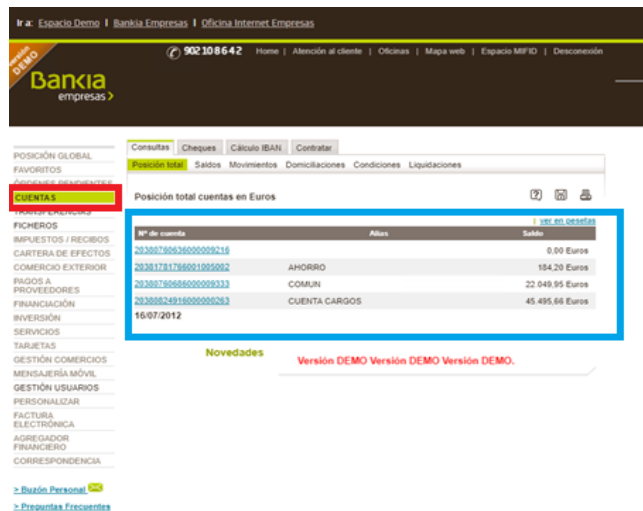
Agregar argumentos (opcional):

Iniciar en (opcional):

Aceptar Cancelar

Resultados del proceso de scrapping

En este punto se explicará como extrae los datos el proceso de scrapping, y que resultados obtenemos de este, utilizando capturas de pantalla de las entidades electrónicas y como esta información visual se relaciona con el código de nuestro proceso. En la imagen que encontramos a continuación, aparece la pantalla principal de la entidad bancaria, en el recuadro de color azul, en el recuadro rojo observamos un botón de menú que nos permite regresar a la página con el listado de cuentas,



Posición total cuentas en Euros

Nº de cuenta	Alias	Saldo
2038076063600009216		0.00 Euros
20381781766001055052	AHORRO	154.20 Euros
2038076066600009333	COMUN	22 049.95 Euros
20380824916000000263	CUENTA CARGOS	45 495.66 Euros
16/07/2012		

Novedades Versión DEMO Versión DEMO Versión DEMO.

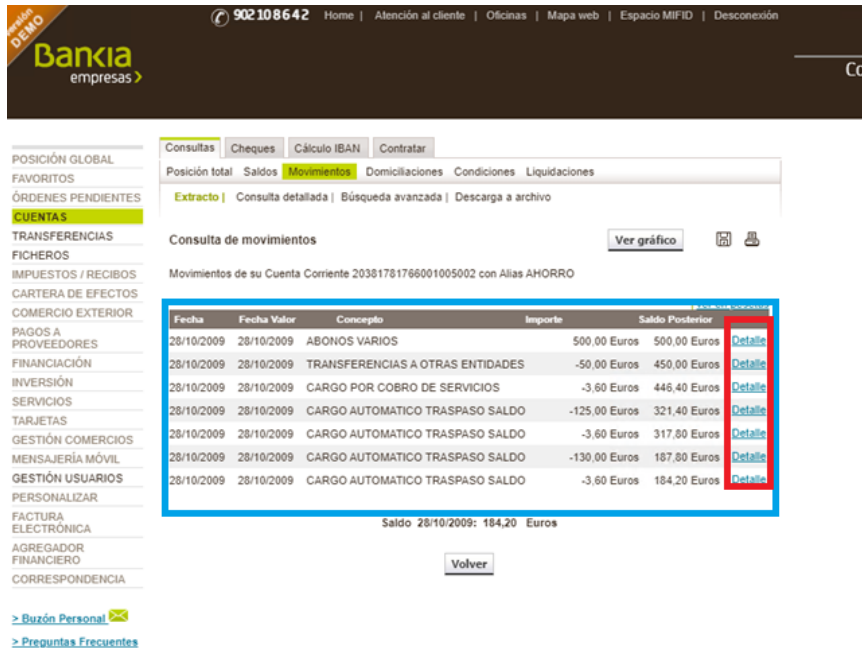
```
<table border="0" cellpadding="0" cellspacing="0" width="595">
  <tbody><tr>
    <td colspan="4">
  </td></tr><tr>
    <td colspan="4" class="verdenormal" align="right" width="595">
      <a href="javascript:void(0);" class="enlaceverdeabajo"
        id="enlacemoneda" nodevalue="ver en euros"
        onclick="conviertePtasEuros()">2038076063600009216</a>
    </td>
  </tr>
</tbody>
</table>
```

En la anterior estructura corresponde a su codificación dentro del código HTML. Este fragmento de la estructura de la etiqueta `<TABLE>`, perteneciente a la tabla de cuentas muestra cómo podemos navegar a través de la estructura como si fueran nodos, pudiendo acceder, como vemos en el ejemplo anterior resaltado, al número de cuenta

De esta forma, podemos localizar este nodo para simular un clic en el momento que precisemos. El ejemplo para acceder a esta ruta sería el siguiente:

```
//a[contains(text(),' CUENTAS ')]
```

```
<a class="selected" id="anchor4" href="tran_puente_oie.htm" onfocus="blur()" >CUENTAS</a>
```



Bankia empresas

902 10 86 42 Home | Atención al cliente | Oficinas | Mapa web | Espacio MIFID | Desconexión

Consultas | Cheques | Cálculo IBAN | Contratar

Posición total | Saldos | **Movimientos** | Domiciliaciones | Condiciones | Liquidaciones

Extracto | Consulta detallada | Búsqueda avanzada | Descarga a archivo

Consulta de movimientos Ver gráfico

Movimientos de su Cuenta Corriente 20381781766001005002 con Alias AHORRO

Fecha	Fecha Valor	Concepto	Importe	Saldo Posterior	Detalle
28/10/2009	28/10/2009	ABONOS VARIOS	500.00 Euros	500.00 Euros	Detalle
28/10/2009	28/10/2009	TRANSFERENCIAS A OTRAS ENTIDADES	-50.00 Euros	450.00 Euros	Detalle
28/10/2009	28/10/2009	CARGO POR COBRO DE SERVICIOS	-3.60 Euros	446.40 Euros	Detalle
28/10/2009	28/10/2009	CARGO AUTOMATICO TRASPASO SALDO	-125.00 Euros	321.40 Euros	Detalle
28/10/2009	28/10/2009	CARGO AUTOMATICO TRASPASO SALDO	-3.60 Euros	317.80 Euros	Detalle
28/10/2009	28/10/2009	CARGO AUTOMATICO TRASPASO SALDO	-130.00 Euros	187.80 Euros	Detalle
28/10/2009	28/10/2009	CARGO AUTOMATICO TRASPASO SALDO	-3.60 Euros	184.20 Euros	Detalle

Saldo 28/10/2009: 184,20 Euros

Volver

> Buzón Personal >>>
> Preguntas Frecuentes

En el ejemplo de la foto anterior, nos encontramos con la página de movimientos de una cuenta en concreto. Resaltado dentro del cuadro azul, tenemos toda la información relativa a los registros actividad que ha tenido nuestra cuenta. La forma de acceder a esta tabla utiliza la misma mecánica que en el caso anterior salvo con una excepción, en el caso de que el concepto del movimiento contenga la palabra transferencia, en el caso concreto de BANKIA, podemos acceder directamente a su detalle utilizando los enlaces situados dentro del rectángulo de color rojo.

```
/**[@id='capaOrigen']/table[1]/tbody/tr[5]/td/table[2]//tbody//tr[" & filas & "]/td[8]
```

Con la ruta anterior, podemos recorrer con un contador “filas”, al enlace del detalle del movimiento, aprovechando a su vez para enlazarlo directamente, por lo que el proceso de categorización no tendrá nada que ejecutar, y de esta forma tenemos más nivel de fiabilidad a la hora de relacionar el movimiento con su detalle.

- Código ejemplo para captura de transferencias (BANKIA)

En esta sección se muestra un ejemplo de la funcionalidad capaz de capturar la información de transferencias y su inserción en base de datos de una banca en concreto:

```
Private Sub capturarDetallTransferencia(idMovimiento As Integer, _movimiento As PB_MMOVIMIENTOS, dr As DriverManage)

    Try

        Dim _TRANSFERENCIA As New PB_MTRANSFERENCIAS

        With _TRANSFERENCIA

            .BENEFICIARIO = dr.getText(By.XPath("//*[@id='capaOrigen']/table[2]/tbody/tr[11]/td[2]/span"))
            .CONCEPTO = dr.getText(By.XPath("//*[@id='capaOrigen']/table[2]/tbody/tr[9]/td[2]/span))
            .CUENTA_BENEFICIARIO = numCuenta
            .CUENTA_ORDENANTE = dr.getText(By.XPath("//*[@id='capaOrigen']/table[2]/tbody/tr[11]/td[2]/span"))
            .IDCUENTA = _movimiento.IDCUENTA
            .GASTOS = dr.getText(By.XPath("//*[@id='capaOrigen']/table[2]/tbody/tr[7]/td[2]"))
            .IDMOVIMIENTO = idMovimiento
            .IDTRANSFERENCIA = 0
            .IMPORTE = dr.getText(By.XPath("//*[@id='capaOrigen']/table[2]/tbody/tr[6]/td[2]"))
            .ORDENANTE = dr.getText(By.XPath("//*[@id='capaOrigen']/table[2]/tbody/tr[10]/td[2]/span"))

        End With

        Capa_SQL.PB_MTRANSFERENCIAS_GRABAR(_TRANSFERENCIA)

        If dr.isDisplayed(btnVolverPaht) Then

            dr.click(btnVolverPaht)

            Thread.Sleep(2000)

        End If

        Catch ex As Exception

        End Try

End Try
```

En la función anterior podemos ver la dinámica que comentábamos, obtenemos cada uno de los datos de la transferencia través de la funcionalidad `getText()`. Obtiene como parámetro adicional el controlador del navegador, para poder hacer uso de sus funcionalidades, y el objeto movimiento que acabamos de capturar e insertar en base de datos, de esta forma, solo tendremos que capturar, utilizando las técnicas anteriores, la información de la transferencia e inicializando el campo `idmovimiento` del objeto `PB_TRANSFERENCIA`, con el campo propio `idmovimiento` del objeto `PB_MOVIMIENTO` que acabamos de insertar en base de datos, proporcionando una alta fiabilidad de la captura del detalle y su relación con el movimiento básico.

Fichero de exportación

```
<RESUMEN>
```

```
<BANCO BANCO="URBALYEM">
```

```
<CUENTA CUENTA="30000001490000222288">
```

```
<MOVIMIENTO>
```

```
<FECHA_OPERACION>07/08/2008 0:00:00</FECHA_OPERACION>
```

```
<FECHA_VALOR>07/08/2008 0:00:00</FECHA_VALOR>
```

```
<CONCPETO>PAGO ALQUILER</CONCPETO>
```

```
<IMPORTE>-50,24</IMPORTE>
```

```
</MOVIMIENTO>
```

```
<MOVIMIENTO>
```

```
<FECHA_OPERACION>07/12/2005 0:00:00</FECHA_OPERACION>
```

```
<FECHA_VALOR>12/12/2005 0:00:00</FECHA_VALOR>
```

```
<CONCPETO>INGRESOCHEQUE</CONCPETO>
```

```
<IMPORTE>3,09</IMPORTE>
```

```
</MOVIMIENTO>
```

```
<MOVIMIENTO>
```

```
<FECHA_OPERACION>02/08/2008 0:00:00</FECHA_OPERACION>
```

```
<FECHA_VALOR>02/08/2008 0:00:00</FECHA_VALOR>
```

```
<CONCPETO>ABONOS VARIOS</CONCPETO>
```

```
<IMPORTE>34,00</IMPORTE>
```

```
<BENEFICIARIO>USUARIO DEMO</BENEFICIARIO>
```

```
<ORDENANTE>ORDENANTE DEMO</ORDENANTE>
```

```
<CUENTA_BENEFICIARIO>30000001490000222288</CUENTA_BENEFICIARIO>
```

```
<CUENTA_ORDENANTE></CUENTA_ORDENANTE>
```

```
<CONCEPTO_TRANSFERENCIA>ABONOS VARIOS</CONCEPTO_TRANSFERENCIA>
```

```
<GASTOS>0,00</GASTOS>
```

```
<IMPORTE_TRANSFERENCIA>34,00</IMPORTE_TRANSFERENCIA>
```

```
</MOVIMIENTO>
```

```
</CUENTA>
```

```
</BANCO>
```

```
</RESUMEN>
```

El fichero de exportación es el resultado final del proceso de scrapping. En este fichero, se encuentra de manera estructurada, todos los datos extraídos en el proceso. El formato de exportación del fichero elegido es en formato XML, se ha optado por este formato ya que es sencillo a la hora de construir y de leer. La composición del fichero se puede ver en la página anterior, con este ejemplo y, diferenciando sus distintas partes en colores, vamos a ir explicando cada una de estas:

1. En primer lugar, encontramos una etiqueta global que va a contener el resto de la estructura, de los datos y de sus atributos, esta etiqueta es la etiqueta `<RESUMEN/>`
2. A partir de la etiqueta `<RESUMEN/>`, encontramos etiquetas que ya contienen información de relevancia. En el orden de la estructura la siguiente etiqueta es `<BANCO/>`, esta etiqueta comprende entre su apertura y su cierre, toda la información referente a una entidad bancaria para ese usuario, es decir, como vemos en el ejemplo de la página anterior, la parte subrayada en color amarillo corresponde con la etiqueta en cuestión, en este caso todo lo que comprenda su etiqueta de inicio y de cierre va indicarnos que toda esa información pertenece al banco Ruralvía, teniendo como atributo de la etiqueta el propio nombre de la entidad
3. Una vez nos encontramos dentro de la etiqueta `<BANCO/>`, inmediatamente tenemos la etiqueta `<CUENTA/>`, esta etiqueta comprende toda la información relativa a un número de cuenta. Como atributo tenemos el número de esta.
4. A partir de esta etiqueta podemos deducir que la siguiente información comprendida dentro de la etiqueta `<CUENTA/>` pertenece a un número de cuenta en concreto, que a su vez pertenece a una entidad bancaria concreta. En este punto nos encontramos etiquetas `<MOVIMIENTO/>`. Esta etiqueta varía según el tipo de movimiento que nos encontramos fruto de la categorización o relación que realiza el proceso de scrapping, sabiendo de esta manera que movimientos son. En las etiquetas de `<MOVIMIENTO/>` sin detalle encontramos una estructura de etiquetas con valores que están asociados a cada uno de los campos de información que nos ofrece la entidad bancaria y que se encuentran en la base de datos del sistema
 - a) `<FECHA_OPERACION/>`
 - b) `<FECHA_VALOR/>`
 - c) `<COCEPTO/>`
 - d) `<IMPORTE/>`

Esta información es la que compone un movimiento básico que ha sido capturado por el proceso de scrapping ofreciendo siempre está mínima información de movimiento bancario. Pero también podemos encontrar un movimiento más detallado. En los casos en el que el movimiento sea un tipo transferencia, el proceso de scrapping previamente habrá realizado su relación o categorización entre el detalle de la transferencia y el movimiento dando como resultado una estructura de movimiento más detallada

- a) <FECHA_OPERACION>
- b) <FECHA_VALOR>
- c) <COCEPTO>
- d) <IMPORTE>
- e) <BENEFICIARIO>
- f) <ORDENANTE>
- g) <CUENTA_BENEFICIARIO>
- h) <CUENTA_ORDENANTE>
- i) <CONCEPTO_TRANSFERENCIA>
- j) <GASTOS>
- k) <IMPORTE>

En cada una de las etiquetas anteriores, encontramos la información referente al detalle del movimiento, aunque en el ámbito de estudio del proyecto solo se abarca el tipo de movimiento de transferencias, es posible utilizando la misma mecánica categorizar un gran porcentaje de los movimientos, pudiendo realizar para cada uno de ellos una estructura dentro de la etiqueta <MOVIMIENTO> con la mínima información, más la parte detallada específica para cada tipo de movimiento bancario.

- **Proceso de generación de fichero XML**

En la solución que se está planteando, se está realizando el proceso de automatización del volcado de datos de entidades financieras con un software empresarial, con el fin de unificar toda la información y procesarlas desde un único sistema. La forma en la que el sistema planteado es capaz de interconectar la banca electrónica con el software de gestión es generando algún tipo de archivo estructurado que sea legible por el sistema de gestión de las empresas y que encapsule toda la información detallada de la banca electrónica, para ello se ha optado en la generación de un archivo XML. En este caso se ha pensado la utilización de este formato por su facilidad a la hora de comprensión visual, pero también porque es un estándar muy extendido y utilizado para muchos sistemas de intercambio de datos por lo que ningún sistema debería de tener problemas para su integración. Podría elegirse cualquier tipo de estándar ya sea por conocimiento del desarrollador en cuestión, por necesidad del cliente final, etc. Para la generación del fichero XML podemos realizarlo de dos formas

1. **Desde la interfaz gráfica** → A través de la opción generar fichero con una cuenta seleccionada ejecutamos la función `generarFichero()` de la clase `GenerarXML`, cuenta como variables públicas `idcuenta` e `idusuario`. Estos parámetros se ejecutan en la función `PB_MMOVIMIENTOS_FICHERO_SELECT` que a su vez pone en funcionamiento el procedimiento almacenado siguiente:

```
WF_PB_MMOVIMIENTOS_TRANSFER_SELECT

@IDMOVIMIENTO AS BIGINT = NULL
,@IDCUENTA AS BIGINT = NULL
,@IDUSUARIO AS BIGINT = NULL
,@FECHA_OPERACION AS SMALLDATETIME = NULL
,@FECHA_VALOR AS SMALLDATETIME = NULL
,@CONCEPTO AS VARCHAR(500) = NULL
,@IMPORTE AS DECIMAL(18,2) = NULL

AS

BEGIN

    SELECT

        T1.[IDMOVIMIENTO]
        ,T1.[IDCUENTA]
        ,T1.[IDUSUARIO]
        ,T3.NOMBRE as 'BANCO'
        ,T2.NUMCUENTA
        ,T1.[FECHA_OPERACION]
        ,T1.[FECHA_VALOR]
        ,T1.[CONCEPTO]
        ,T1.[IMPORTE]
        ,T4.BENEFICIARIO
        ,T4.ORDENANTE
        ,T4.CUENTA_BENEFICIARIO
        ,T4.CUENTA_ORDENANTE
        ,T4.CONCEPTO AS 'CONCEPTO_TRANSFERENCIA'
        ,T4.GASTOS
        ,T4.IMPORTE AS 'IMPORTE_TRANSFERENCIA'
    FROM [dbo].[PB_MMOVIMIENTOS] T1
    LEFT JOIN PB_MCUENTAS T2 ON T1.IDCUENTA = T2.IDCUENTA
    LEFT JOIN PB_MBANCOS T3 ON T2.IDBANCO = T3.IDBANCO
    LEFT JOIN PB_MTRANSFERENCIAS T4 ON T1.IDMOVIMIENTO = T4.IDMOVIMIENTO
    WHERE

        (T1.IDMOVIMIENTO = @IDMOVIMIENTO OR @IDMOVIMIENTO IS NULL) AND
        (T1.IDCUENTA = @IDCUENTA OR @IDCUENTA IS NULL) AND
        (T1.IDUSUARIO = @IDUSUARIO OR @IDUSUARIO IS NULL) AND
        (T1.FECHA_OPERACION = @FECHA_OPERACION OR @FECHA_OPERACION IS NULL) AND
        (T1.FECHA_VALOR = @FECHA_VALOR OR @FECHA_VALOR IS NULL) AND
        (T1.CONCEPTO = @CONCEPTO OR @CONCEPTO IS NULL) AND
        (T1.IMPORTE = @IMPORTE OR @IMPORTE IS NULL)
```

En este procedimiento realizamos un cruce entre tablas, por un lado, realizamos una sentencia `SELECT` contra la tabla `PB_MOVIMIENTOS`, en esta tabla encontramos toda la información mínima de cada uno de los movimientos. Para poder relacionar los datos, tenemos que cruzar esta tabla con `PB_TRANSFERENCIAS` principalmente, de esta manera utilizamos como factor para relacionarlas el campo `IDMOVIMIENTO`, ya que es el campo que identifica un único registro en la tabla `PB_MMOVIMIENTO`, por lo tanto, cruzará todos los datos y añadirá, con valores o no, en el caso de que el movimiento no sea una transferencia y por lo tanto no tenga información detallada. Por último, cruzamos también con las tablas `PB_MCUENTAS` y `PB_MBANCOS` utilizando con factor de relación los campos `idcuenta` y `idbanco` para obtener el número de cuenta y el nombre del banco ya que solo contamos con los identificadores de cada uno de ellos y no con sus nombres literales para la posterior presentación de la información. Los demás parámetros del procedimiento están pensados para poder tener un filtrado más exhaustivo, pero en este caso no son necesarios por lo tanto se ignoran dentro de la cláusula `WHERE`.

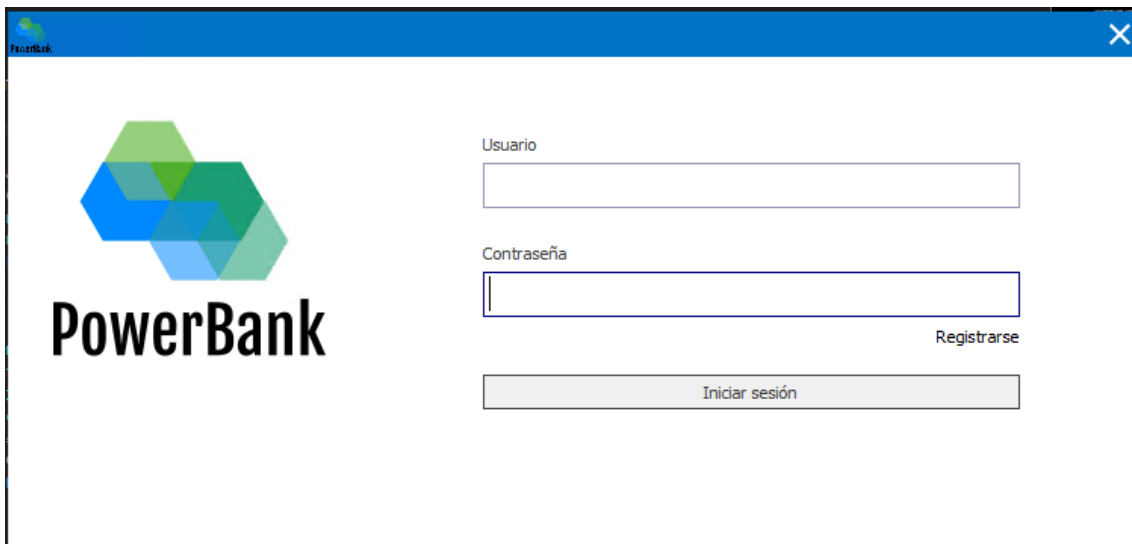
- 2. Automáticamente en el proceso de scrapping** → Este sería el proceso automático en el que se basa el modelo de negocio, una vez el proceso de scrapping termina con la categorización, ejecuta exactamente la misma clase anterior con la diferencia de los parámetros que introducimos ya que serán los específicos para cada cliente y acceso. Como observamos, anteriormente encontrábamos ciertos parámetros del procedimiento almacenado que se ignoraban dentro de la cláusula `WHERE` y que ahora serán utilizados, en este caso, los parámetros introducidos además del `idusuario`, también se utilizarán la fecha de operación como filtrado, ya que solo queremos los datos que han sido capturados en el último día con el fin de no sobrecargar el fichero y que la integración con un posible sistema de información se más rápida y liviana, y por otro lado también reducimos la posibilidad de errores. En este caso la ruta de generación del fichero se escoge directamente en el escritorio del propio usuario, pero esta ruta es totalmente configurable con el objetivo de generarlo dentro de un directorio con conexión FTP para tener una base común con el sistema de información que tenga que integrar esa información.

Interfaz Gráfica

En este punto hablaremos sobre el desarrollo de la interfaz gráfica y de las distintas funcionalidades que puedo realizar desde esta, así como la información que puedo consultar.

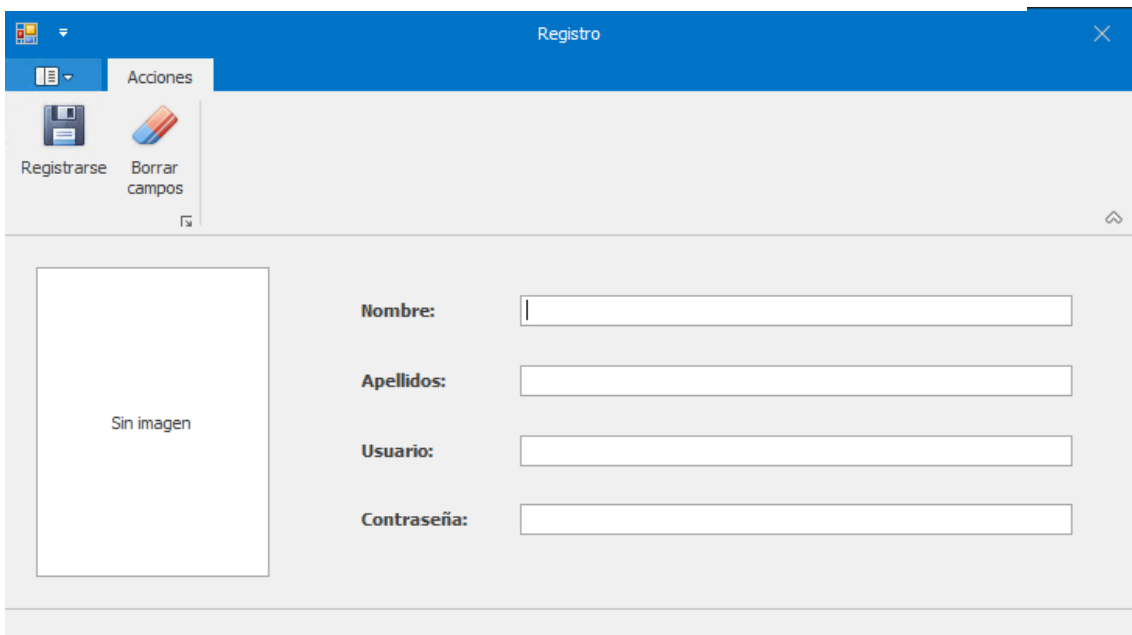
Pantalla de Login y formulario de registro

Formulario de inicio de sesión para poder acceder a la aplicación o registrarse en esta



The screenshot shows a web browser window titled "PowerBank". On the left side, there is a logo consisting of several overlapping hexagons in shades of blue and green, with the text "PowerBank" below it. On the right side, there is a login form with two input fields: "Usuario" and "Contraseña". Below the "Contraseña" field, there is a "Registrarse" link. At the bottom of the form, there is a "Iniciar sesión" button.

Formulario de inicio de sesión



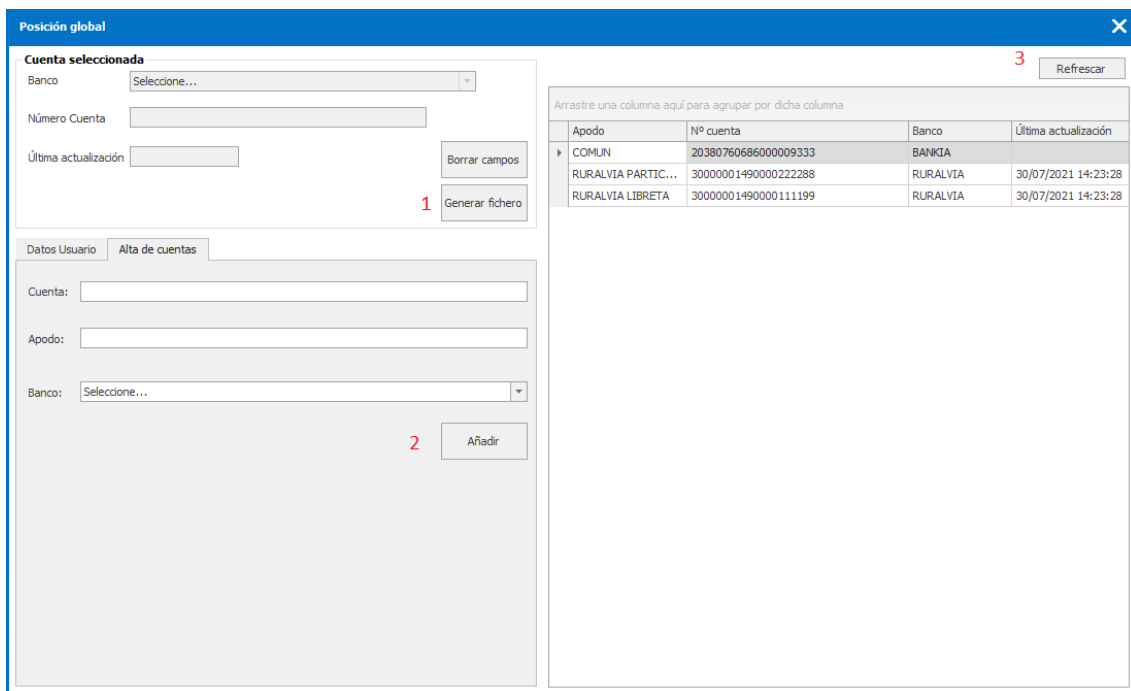
The screenshot shows a web browser window titled "Registro". At the top, there is a blue header with the title "Registro" and a close button. Below the header, there is a toolbar with two buttons: "Registrarse" and "Borrar campos". The main content area contains a registration form with four input fields: "Nombre:", "Apellidos:", "Usuario:", and "Contraseña:". To the left of the form, there is a placeholder for a profile picture with the text "Sin imagen".

Formulario de registro sencillo para poder acceder a la aplicación

Menú principal

La pantalla de menú principal ofrece distintos tipos de información y acciones. La información que nos proporciona la interfaz es sencilla, en la parte inferior izquierda podemos obtener en la pestaña de datos de usuario la información referente a este dentro del sistema, también encontramos una segunda pestaña de alta de cuentas, en este apartado podemos dar de alta una cuenta nueva que se va a introducir dentro del proceso de scrapping y poder obtener la información sobre esta. En la parte derecha de la interfaz podemos obtener la información de todas las cuentas que tenemos dentro del proceso de scrapping, así como la información de la última actualización de la información de cada una de las cuentas. Por último, vamos a comentar las tres acciones que se pueden realizar:

- 1- **Generar fichero** → Esta funcionalidad permite generar en el equipo del usuario el fichero XML con los datos referente a una cuenta en concreto que seleccione realizando doble clic sobre la tabla de cuentas.
- 2- **Añadir** → Botón para insertar una nueva cuenta al proceso.
- 3- **Refrescar** → Permite actualizar la información de última actualización de cada cuenta para obtener el último momento en el que se incorporaron los datos de cada una de ellas.



Apodo	Nº cuenta	Banco	Última actualización
COMUN	2038076068600009333	BANKIA	
RURALVIA PARTIC...	30000001490000222288	RURALVIA	30/07/2021 14:23:28
RURALVIA LIBRETA	30000001490000111199	RURALVIA	30/07/2021 14:23:28

Formulario de posición global de la aplicación

Resultados

Según todas las pruebas realizadas durante y posteriormente a el desarrollo de la herramienta, se han obtenido los siguientes resultados:

- Se ha conseguido acceder de manera eficiente a las entidades bancarias de pruebas, así como a la información de estas, pudiendo ser integradas dentro de la base de datos del sistema.
- La velocidad de extracción, almacenamiento y procesamiento de la información es mucho mayor que la de un usuario físico, realizando el proceso completo de scrapping en tiempos inferiores a los tres minutos por acceso.
- La categorización de la información no produce situaciones de inestabilidad de los datos por lo que los datos resultantes están consolidados.
- La exportación de la información a un formato estándar permite la integración de una gran cantidad de movimientos en poco tiempo.
- El proceso de scrapping está sometido a una mejora continua ya que no depende de algo que el sistema pueda controlar, si no que depende de un agente externo, en este caso la entidad bancaria.

Relación con el grado

Como hemos visto a lo largo del documento, el proyecto cuenta con dos partes importantes, una más tecnológica y muy vinculada con técnicas de programación, y la parte más enfocada al negocio. En este proyecto se realiza una interacción entre estas dos partes en la que vemos como un proyecto tecnológico enfocado a necesidades de automatización de procesos de la empresa ayuda a mejorar el rendimiento de esta. El grado de sistemas de información se enfoca precisamente en estos dos ámbitos que comentamos, por un lado, el grado se centra en la formación complementaria entre tecnologías de la información, y organización de empresas. Los sistemas de información se centran en la aplicación de estas tecnologías en las empresas, también está enfocada en la comprensión y descripción relacionados con el uso y explotación de la información en las organizaciones lo que adquiere una relación directa con el objetivo del trabajo. También cabe indicar sobre el grado que, al ser una rama de ingeniería informática, está muy ligado también con el uso de tecnologías y de la programación para crear rutinas y procesos de automatización, así como el mantenimiento y escalabilidad. Por último, se ponen en práctica objetivos claros en relación con el grado:

1. Análisis del sistema
2. Control y gestión del proyecto
3. Creación y gestión de modelos de datos
4. Diseño de sistemas
5. Conocimiento de la organización de las empresas
6. Diseño de modelo de negocio

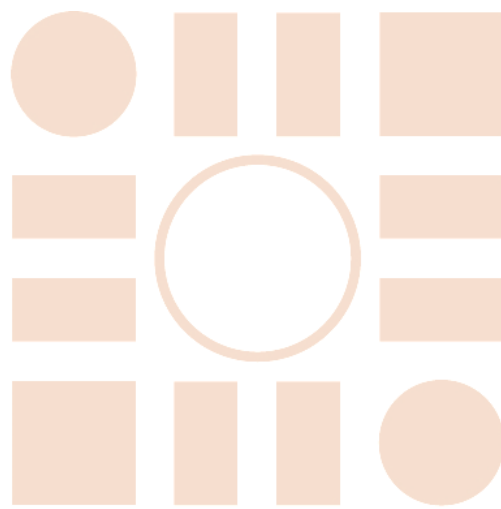
Conclusiones

En este proyecto se ha analizado como un proceso no automatizado dentro de una empresa puede ser costoso, y que la inversión en ciertas ocasiones no es muy alta a la hora de implementar tecnologías que a largo plazo ahorra costes a esta. En este proyecto se ha adquirido una comprensión sobre los problemas reales de las empresas a la hora de automatizar procesos o de ser un paso más eficiente en tareas que se están realizando dentro de las organizaciones. Este aprendizaje puede dar un punto de vista de que evidentemente las empresas todavía tienen muchas necesidades de automatización en muchos de sus sectores. Como hemos podido comprobar, los resultados que se arrojan desde el trabajo son realmente prometedores, logrando realizar la generación de un tipo de fichero con información valiosa, con una infraestructura prácticamente sin costes, y que por otra parte puede ahorrar tiempo y dinero a las empresas. No obstante, la versión del sistema ofrecido en este proyecto, lo podríamos denominar con una base para futuras ampliaciones, el poder obtener toda la información que requiera el usuario, no solo con el uso de procesos scrapping, si no también apoyándose en APIS realizadas por entidades financieras que puedan agilizar el proceso de scrapping, así como la recopilación de más tipos de información que pueden ofrecerse al usuario final. Se han analizado también las posibles debilidades del sistema, así como los son cambios en las bancas, desconexiones de red entre otras, pero todas estas casuísticas pueden solucionarse de manera rápida y eficiente. Para concluir, podemos resumir en que la solución aportada genera valor para el cliente o el usuario final, aportando una información consolidada, que ahorra capital humano a las empresas, gana en eficiencia y procesamiento de datos y como ya hemos comentado ayuda potencialmente a la toma de decisiones ya que la información generada en el proceso puede ser comprendida por los sistemas de información actuales

Bibliografía

- [1] Departamento de investigación del IEB, “ANUARIO IEB DE BANCA DIGITAL Y FINTECH 2017,” 2017.
- [2] J. Tortajada, “Análisis del sector financiero: Nuevas tecnologías y modelos de negocio,” 2018.
- [3] Isabel Ballester, “Los sistemas ERP y la importancia de la contabilidad analítica,” p. 4, 2008.
- [4] M. Duranza *et al.*, “Nueva directiva PSD2 y sus efectos sobre el sistema financiero español,” Tenerife, 2016.
- [5] R. C. Montenegro, “Testing y calidad de software. automatización de pruebas con selenium webdriver,” *Cadime*, p. 134, 2017, [Online]. Available: http://oa.upm.es/49320/1/PFC_RAFAEL_CUBAS_MONTENEGRO.pdf.
- [6] “XPath - Wikipedia, la enciclopedia libre.” [https://es.wikipedia.org/wiki/XPath#Introducción](https://es.wikipedia.org/wiki/XPath#Introducci3n) (accessed Oct. 02, 2021).
- [7] I. E. Marini, “El Modelo Cliente / Servidor,” pp. 1–11, 2012, [Online]. Available: <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>.

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá