**VISTAS ARQUITECTÓNICAS Y DESCRIPCIÓN DE LA PLATAFORMA**

**Equipo de Trabajo**

Arley Berrio Arroyave

Yenifer Daniela Osorio Gómez

Carlos Alberto Medina Castillo

Jhonatan Alejandro Torres Vasco

**Asesor**

Diego Botia Valderrama

**Historia**

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Descripción de Cambios** |
| 1.0 | 06/06/2020 | Inicio |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Descripción del Módulo:**

Al tener dificultades de poner procesos delicados del negocio en manos del personal humano se pueden tener varios riesgos, uno de eso es el perder información, y más específicamente en el caso del problema que nos concierne es posible perder dinero, ya que solo personal relacionado directamente del banco podrá hacer la transferencia de dinero que las empresas le pidan, manejando millones y realizando un proceso que puede ser confuso para personas que no lo manejan constantemente. Es por eso que se ve la importancia de automatizar el proceso usando Robotic Process Automation, y al módulo web existente se le realizará un análisis de datos del proceso de transacción que es el que presenta las dificultades antes expuestas y se identificarán fallas del sistema a partir de técnicas de minería de procesos.

**Requisitos funcionales:**

* El sistema debe notificar en caso de cualquier anomalía y continuar con el proceso de transacción.
* Identificar cuellos de botella y fallas en el proceso de negocio utilizando minería de datos.
* El sistema debe permitir el registro de transacciones.
* El sistema deberá validar los montos de transacciones para solicitar información adicional.
* El sistema debe permitir anexar información adicional a las transacciones que así lo requieran.
* El sistema deberá notificar las transacciones que cuenten con los anexos requeridos como fraudulentas.
* El sistema deberá permitir transacciones entre diferentes empresas.
* El sistema debe generar logs, para facilitar el seguimiento del bot.

**Requisitos no funcionales:**

* Es un entorno productivo; el bot no puede detenerse debido a errores del usuario.
* Se optimizará en la plataforma web.
* Se realizará utilizando las siguientes tecnologías: Uipath, Uipath Orchestrator, Prom, Openrefine.
* Optimizar el proceso de transacciones.
* Los logs generados por el bot deben poder visualizarse en una vista en una página web.
* Debe poder realizar miles de transacciones bancarias, sin ver afectada la estabilidad del sistema.

**Descripción de los Roles del Equipo**

**Daniela Osorio:** BackEnd**,** Arquitecta. Encargada de presentar arquitecturas candidatas y del manejo de las tecnologías escogidas.

**Carlos Medina:** Testing **-** FrontEnd, encargado de diseñar las vistas para la visualización de los reportes generados por la aplicación. Y encargado de realizar testeo de las diferentes funcionalidades del bot, mediante el uso de pruebas unitarias, despliegue, funcionales.

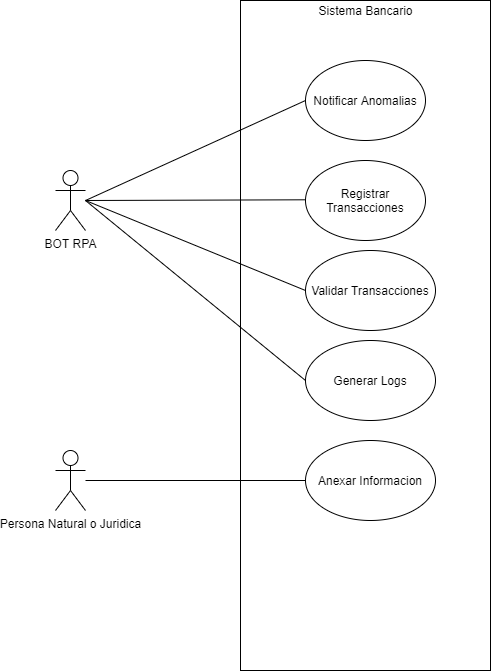
**Jhonatan Torres Vasco:**  BackEnd, Analista. Encargado del trabajo con Uipath, y de analizar el problema y plantearlo en forma de casos de uso, e historias de usuario.

**Arley Berrio:** Líder de proyecto, Backend, Arquitecto lider. encargado de

escoger las tecnologías y frameworks a usar y la arquitectura a seguir; asignación de tareas para los demás miembros y una participación activa en el Backend.

**Descripción del Proceso:**

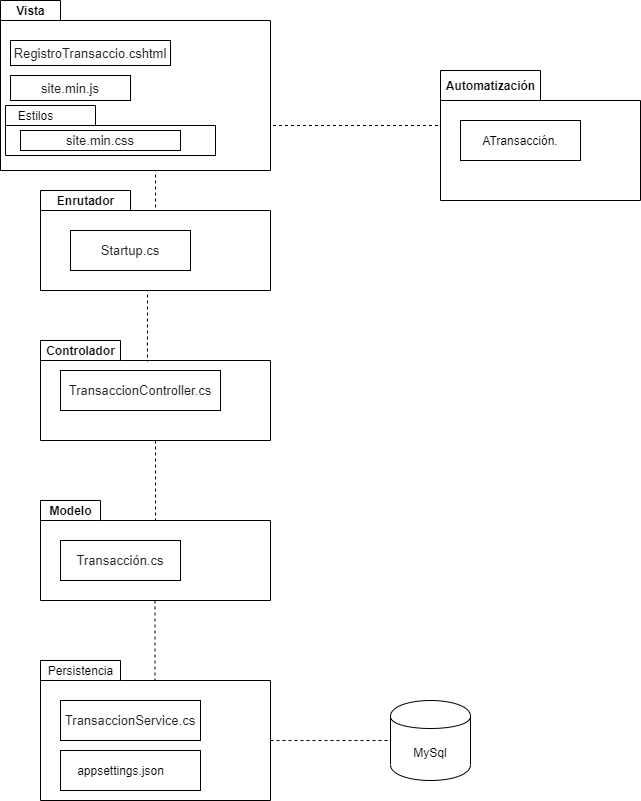
Teniendo como proceso crítico las transacciones, y como una de las soluciones es automatizar este proceso, utilizamos la herramienta de Uipath,con la cual se usará robots de RPA los cuales obtendrán archivos .csv que contendrán las transacciones previamente almacenadas en una base de datos, y con los datos que toma rellena formularios de nuevo uno por uno, para así almacenarlos en una nueva base de datos, a su vez analiza los datos para encontrar anomalías si es el caso,y posteriormente los notificará.

**Vista Conceptual**

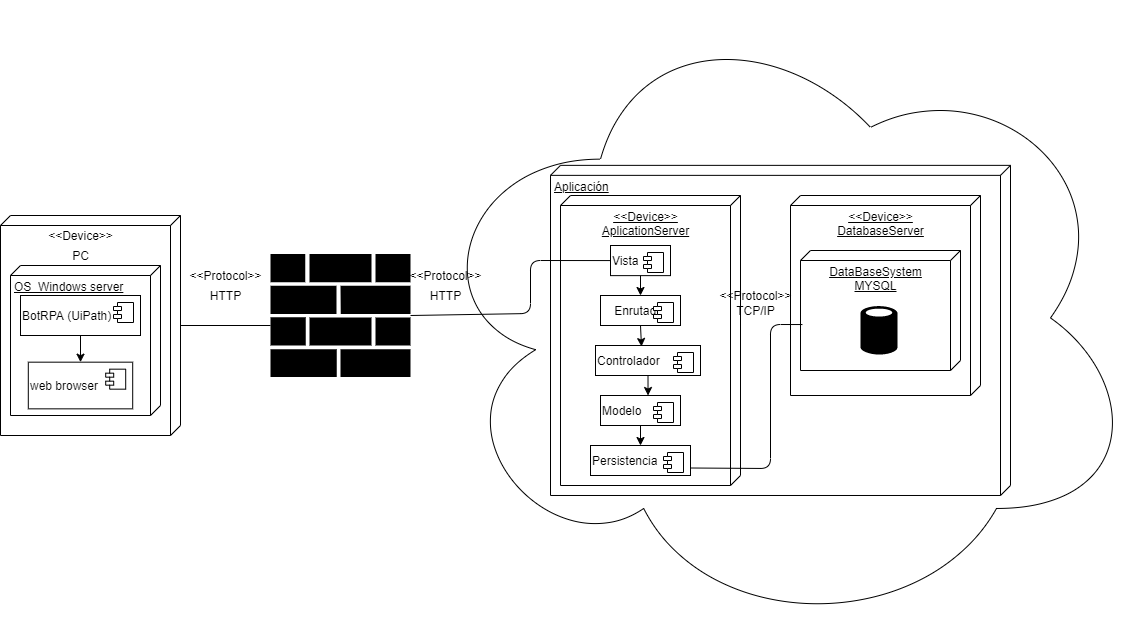
|  |
| --- |
| **HU- 0001 Anexar información**  **Como**: Persona natural o juridica  **Quiero**: Anexar información al sistema con respecto a la transacción que se va a llevar a cabo  **Para**: El sistema tenga información que registrar |
| **CRITERIOS DE ACEPTACIÓN**  **Cuando**: La información que se ingresa en el formulario no es completa  **Espero**   1. Notificar que la información es incompleta 2. No permitir enviar formulario   **Cuando:** La información ingresada en los espacios con es válida, es decir, tipo de dato incorrecto  **Espero**   1. Notificar que el tipo de dato no concuerda 2. No permitir enviar formulario |

|  |
| --- |
| **HU- 0002 Registrar transacción**  **Como**: Banco  **Quiero**: Registrar transacciones  **Para**: Mantener la integridad y fiabilidad de los datos, reduciendo al máximo la intervención humana. |
| **CRITERIOS DE ACEPTACIÓN**  **Cuando**: Se realiza una transacción  **Espero**   1. Tipo de transacción 2. El monto de la transacción 3. Cuenta de origen y destino 4. Tipo de cuentas   **Cuando:** El monto de la transacción sea mayor a 10 millones de pesos.  **Espero**   1. Tipo de transacción 2. El monto de la transacción 3. Cuenta de origen y destino 4. Tipo de cuenta 5. Anexar firma de propietario de cuenta de origen. |

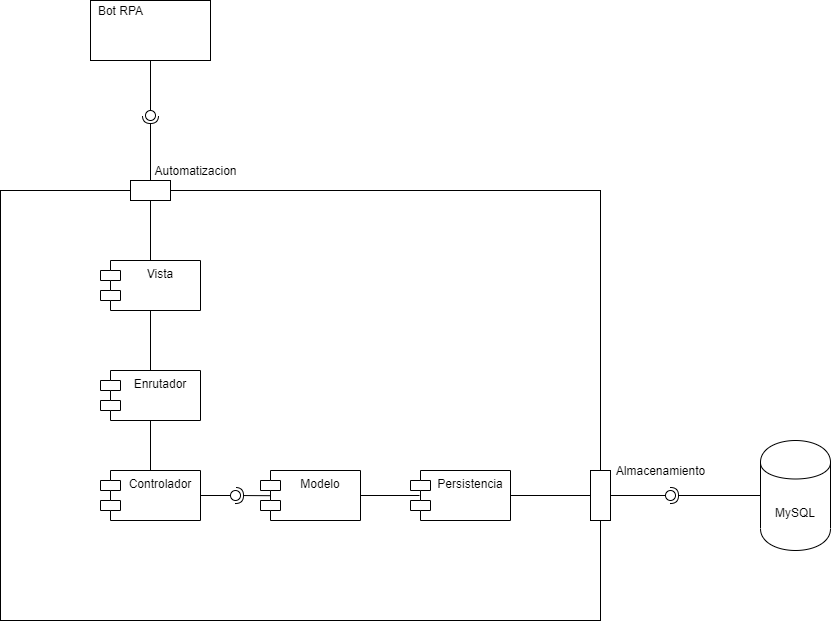
**Vista Lógica**



**Vista Física**



**Vista de Implementación**

****

**Evaluación de Plataformas y Frameworks de Desarrollo**

**1. Fase 1: Selección de la Plataforma**

* JEE (Java Enterprise Edition): Es una plataforma de programación para desarrollar y ejecutar software de aplicaciones en el lenguaje de programación java, el cual permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

Java EE tiene varias especificaciones de API, tales como JDBC, RMI, JMS, Servicios Web, XML, etc. además define como coordinarlas, su principal funcionalidad es utilizarlo en desarrollos de aplicaciones empresariales, debido a que comprenden un conjunto de especificaciones y funcionalidades orientadas a negocios. Entre sus ventajas se encuentran la gran robustez, fiabilidad, estabilidad y seguridad.

# Ventajas

* Simplifica la complejidad de construir aplicaciones n-tier
* Permite lidiar con fuertes requerimientos no funcionales
* Tolerancia a fallas
* Estandariza una API común entre componentes y servidores de aplicación
* Los servidores de aplicación y contenedores proveen los servicios del framework.

# Desventajas

* Balas de plata
* Buzzword Compliant
* Falta de profesionales debidamente capacitados
* **ASP .NET CORE:** es un framework de código abierto y multiplataforma para la creación de aplicaciones modernas conectadas a Internet, como aplicaciones web y APIs Web. Se diseñó para proporcionar un framework de desarrollo optimizado para las aplicaciones que se implementan tanto en la nube como en servidores dedicados en las instalaciones del cliente.

Se pueden desarrollar y ejecutar aplicaciones ASP.NET Core en Windows, Mac y Linux.

**¿ Cuando se debe usar ?**

Cuando queramos aplicaciones multiplataformas,

* Para mejorar la escalabilidad de nuestras aplicaciones, en si esto no es un feature. Pero con NET Core todo se vuelve más escalable.
* Tener como objetivo el desarrollo de microservicios.
* Queramos trabajar con contenedores de Docker.

NET Core su arquitectura permite escalar mejor y tiene un mejor performance (más rápido).

* **MEAN (Mongo / Express / Angular / Node):**

También llamado Stack MEAN. Posibilitó crear aplicaciones de despliegue, usando JavaScript en cada una de las fases y capas del sistema: al cliente, al servidor y al almacenamiento. Sus partes principales son:

**Node js (para el lado del servidor):**

Es un entorno de ejecución de aplicaciones multiplataforma y de código abierto (Open Source). Utiliza por debajo el motor de JavaScript de Google (V8), tiene una arquitectura orientada a evento, que le proporcionan un rendimiento y una escalabilidad muy elevadas. Se puede utilizar para crear cualquier tipo de lógica de aplicación, pero dado que incorpora un módulo para poder actuar como un servidor web, es especialmente popular para crear aplicaciones web.

**Express:**

Es un framework que está escrito en JavaScript para Node.js. Su objetivo es ofrecer soporte para las principales necesidades en este tipo de aplicaciones: gestión de peticiones y respuestas, cabeceras, rutas, vistas, cookies, etcétera.

**Angular (para el lado del cliente):**

Es un framework opensource desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página.

Angular separa el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador).

**Mongo DB(Para el lado de almacenamiento):**

Es un sistema de base de datos noSQL de código abierto. En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (Binary JSON), los cuales utilizan JavaScript para representar la información, recibiendo, enviando y almacenando datos usando la notación JSON con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. Actualmente, los tipos de información que suelen requerir las aplicaciones web demandan mayor flexibilidad, menos coherencia y sobre todo mayor capacidad de escalar. Para dar respuesta a todo esto surge la tendencia tecnológica en almacenes de datos que se denomina NoSQL.

**Servidor de aplicaciones WildFly (Antes conocido como JBoss ):**

Es un servidor de aplicaciones muy utilizado, debido a sus múltiples ventajas entre ellas podemos resaltar las siguientes:

* Al ser de código abierto, se nutre constantemente por su red de colaboradores,.
* Inicia rápido de este servidor, tal como se describe en su sirio web oficial:

*“En el proceso de arranque altamente optimizado de WildFly 8, los servicios se inician simultáneamente para eliminar esperas innecesarias y aprovechar el poder de los procesadores multi-core. Los servicios no críticos se mantienen en hielo hasta el primer uso. “*

* Cuenta con una parte integral llamada Undertow el cual es un servidor web que tiene la capacidad de escalar a más de un millón de conexiones.
* Es multiplataforma.
  1. **Selección de las plataformas**.

Describir brevemente cada plataforma. Se recomienda hacer un completo cuadro comparativo entre estas.

* 1. **Selección de las características a evaluar.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **JavaEnterprise Edition(JEE)** | **Node js** | **.Net core** |
| Documentación | JEE cuenta con amplia documentación para cada una de sus versiones en la página de oracle, en la cual se pueden encontrar guías de varios tipos como de instalación, de apis y tutoriales. | Nodejs cuenta con gran cantidad de documentación en su página nodejs.org. Dicha documentación tiene información detallada sobre su función y objeto, y guías extensas sobre las características y capacidades técnicas de sus versiones | Cuenta con una amplia y completa documentación en la web de docs.microsoft, basado en el lenguaje de programación c#, .net core cuenta con el respaldo de Microsoft, lo que permite encontrar fácilmente información que va desde lo más básico hasta niveles avanzados. |
| Seguridad | JEE cuenta con diferentes métodos para garantizar la seguridad desde la PKI (infraestructura de clave pública) incluyendo la autenticación con certificados, hasta el servicio JAAS de autenticacion y autorizacion para código en el servidor de aplicaciones y la transmisión segura. | Nodejs tiene un equipo encargado de evaluar los problemas de seguridad encontrados por ellos mismos o por la comunidad que lo soporta. Los reportes  de seguridad son  enviados vía  HackerOne por el  equipo, los cuales  tienen un tiempo  estimado donde le  envían a los diferentes  usuarios los pasos para  la solución del bug  encontrado. | ASP.NET Core permite a los desarrolladores configurar y administrar fácilmente la seguridad de sus aplicaciones. ASP.NET Core contiene características para administrar la autenticación, la autorización, la protección de datos, la aplicación de HTTPS, los secretos de la aplicación, la protección contra falsificaciones y la administración de CORS. Estas características de seguridad le permiten crear aplicaciones ASP.NET Core robustas pero seguras. |
| Facilidad de uso | Java cuenta con una gran variedad de IDEs tales como Netbeans, Eclipse, BlueJ entre otros, los cuales son sencillos de instalar y además es compatibles con diversos sistemas operativos y soporta una gran cantidad de frameworks.  Sus IDEs cuentan con una gran variedad de herramientas que proporcionan una gran facilidad en el momento de administrar y crear un proyecto tanto web, de escritorio, servidor, entre otros. | Debido a su comunidad, NodeJs cuenta con diversos instructivos de descarga, instalación y adaptación de gran variedad de frameworks, además, la instalación puede ser usando gestor de  paquetes, así es más fácil su implementación para el usuario. Es soportado por los principales sistemas operativos y en su página principal se pueden obtener todas sus versiones con las respectivas especificaciones. | C # se puede usar para una gran variedad de propósitos, desde el desarrollo web hasta las aplicaciones de consola y gracias a la plataforma .NET se puede crear prácticamente de todo: apps de escritorio, servidores, cloud, móviles...  La sintaxis de C# se base en C++ (y en Java), por lo que a priori podría parecer un lenguaje complejo para principiantes. Sin embargo, las opciones de autocompletado de Visual Studio, la auto-creación de proyectos y la facilidad de uso de su entorno de desarrollo en general, son aspectos que hacen que este lenguaje no sea difícil de aprender. |
| Internacionalización | JEE maneja los diferentes idiomas mediante archivos .properties creando un archivo por cada idioma que se quiere tener cambiando solo los valores por cada idioma que se tenga. | NodeJS, cuenta con un módulo completo sobre internacionalización  donde muestra todas las funciones correspondientes y cómo implementarlas. | Soporte para Globalización de .NET  Contiene clases que definen información relativa a:  - Idioma (e.g. castellano vs inglés)  - País o región (e.g. España vs México)  - Calendarios utilizados (e.g. gregoriano)  - Modelos de formato para las fechas (e.g. yy/mm/dd vs.  dd/mm/yy)  - Formato numérico (e.g. 12,000.00 vs 12.000,00)  - Moneda (e.g. €)  - Etc. |
| Manejo de errores | JEE cuenta con un manejo de excepciones el cual crea un objeto que contiene información sobre la excepción y retorna en forma inusual al código  llamador con la información de la excepción, a esta forma de encapsular la excepción se le conoce como catch, como estas excepciones son identificables podemos realizar un tratamiento especial para ellas y ejecutar comandos que aseguren el correcto funcionamiento de nuestra aplicación. | Las aplicaciones ejecutadas en NodeJS, normalmente experimentan cuatro tipos de errores: errores estándares de JavaScript, Errores del sistema provocados por restricciones subyacentes del sistema operativo,  errores especificados por el usuario activados por el código de la aplicación y errores de afirmación. Todos estos errores heredan de la clase <Error> estándar de JavaScript. NodeJS tiene un modulo para manejo de errores en su documentación. | Para el manejo de errores cuenta con el bien conocido  try - catch - finally, además de los throw - throws, permitiendo la personalización de los errores y cómo capturarlos y lanzarlos.  dentro de la documentación se encuentran explicados más detalladamente |
| Soporte a nuevos estándares | Oracle se encarga de mantener actualizado las tecnologías de JEE para que no solo sigan siendo compatibles con las nuevas tecnologías si no que logren contribuir para aumentar los beneficios que estas nos ofrecen. | Debido a que NodeJS es un entorno de  desarrollo  multiplataforma  relativamente nuevo, enfocado en programas  de red altamente  escalables como son los servidores web, tiene buen soporte a los nuevos estándares. Además, como se le hacen constantes  actualizaciones,  permite tener una  mejor adaptabilidad en  las nuevas tendencias  tecnológicas. | Desde Microsoft regularmente se realizan actualizaciones del marco .NET, estando siempre a la par de las necesidades del mercado, desarrollándose en el lenguaje de programación C#, admite desarrollo de Aplicaciones Web, empleando Web services, conexiones a proveedores externos, APIS.. entre otras tecnologías, incluyendo además el desarrollo de aplicaciones de escritorio y móviles con xamarin. |
| Mecanismos de Licenciamiento | Está el openjdk y  la JVM de Oracle los cuales son open source, y el JVM de IBM el cual es de pago. | NodeJS cuenta con la  licencia MIT que es una licencia de software libre permisiva, compatible con DFSG (directrices de software libre de  Debian) y GPL (General  Public License ), aprobado por la FSF  (free software foundation) y la OSI  (open source initiative) | El marco .NET Core es de [código abierto](https://github.com/dotnet/core), con licencias de MIT y Apache 2. .NET Core es un proyecto de [.NET Foundation](https://dotnetfoundation.org/). |

* Documentación: Manuales, ejemplos y comunidad de desarrollo del framework para garantizar un adecuado soporte.
* Seguridad: Manejo de usuarios y mecanismos de personalización de menús para la adecuada implementación de roles.
* Facilidad de uso: Proceso de instalación sencillo, al igual que las actualizaciones y adaptaciones sobre el framework.
* Internacionalización: mecanismos para el manejo de parámetros que permitan la internacionalización del portal.
* Manejo de errores: Captura y personalización de los errores para un adecuado tratamiento y presentación de los mismos.
* Soporte a nuevos estándares de la Web 2.0 o Web 3.0. Mashups, Tagging, XML, Web Services, Cloud Computing, Redes Sociales, Portlets, Soporte a Aplicaciones Moviles en plataformas como Android de Google o iOS de Apple.
* Mecanismos de Licenciamiento. Evaluar el tipo de licencias, ventajas de tener software libre o propietario, interconexión con diferentes DBMS, utilización en empresas desarrolladoras de software.

1.3. Valoración de las características.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **JavaEnterprise Edition(JEE)** | **Node js** | **.Net core** |
| Documentación | 9 | 9 | 9 |
| Seguridad | 9 | 8 | 8 |
| Facilidad de uso | 8 | 7 | 9 |
| Internacionalización | 8 | 9 | 9 |
| Manejo de errores | 8 | 8 | 8 |
| Soporte a nuevos  estándares | 8 | 8 | 9 |
| Mecanismos de  Licenciamiento | 9 | 9 | 9 |

1.4. Calificación de cada plataforma.

|  |  |
| --- | --- |
|  | **Total** |
| **JavaEnterprise Edition(JEE)** | 8.4 |
| **Node js** | 8.3 |
| .**Net core** | 8.7 |

1.5. Cómputo de totales y selección de la mejor plataforma.

Al evaluar todos los criterios y promediar las calificaciones que como grupo asignamos a cada una de las opciones elegidas, podemos observar a .Net core como el mayor candidato, en esta tecnología se tenían algunas nociones básicas, pero en el transcurso de la investigación para asignar las puntuaciones vimos que no era muy compleja de aprender y podría ser una buena oportunidad para aprender nuevas cosas.

**2. Fase 2: Selección del Framework**

2.1. Selección de los frameworks.

**Bootstrap**: Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario, es open source y actualmente se encuentra en la versión 4.4, este framework combina CSS y JavaScript para estilizar los elementos de una página HTML, ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

2.2. Selección de las características a evaluar.

|  |  |  |
| --- | --- | --- |
|  | **Bootstrap** | **Angular** |
| Documentación | Cuenta con una información muy detallada de sus hojas de estilo, además de imágenes que muestran su visualización, especificaciones de sus clases y sus identificadores. | Este framework cuenta con amplia información con respecto al uso de este y su plataforma de desarrollo, desde su primera aplicación hasta la optimización de aplicaciones complejas de una sola página para empresas. Los tutoriales y guías incluyen ejemplos descargables para acelerar sus proyectos. |
| Seguridad | Bootstrap cuenta con validaciones para los campos. | La vigilancia de los datos y la información en tiempo real al usuario son dos pilares de Angular. Angular cuenta con documentación la cual describe las protecciones integradas de Angular contra vulnerabilidades y ataques comunes de aplicaciones web, como los ataques de secuencias de comandos entre sitios. |
| Facilidad de uso | Bootstrap es muy intuitivo, se adaptada a cualquier programador la cual permite crear grandes sitios web en poco tiempo, además Bootstrap nos provee de todas las reglas CSS para que nuestro sitio web se adapte de manera dinámica a la mayoría de las pantallas. | Angular tiene una gran comunidad, por esta razón es muy fácil encontrar información con respecto a su uso y a la solución de problemas comunes. |
| Internacionalización | Bootstrap cuenta con documentación en varios idiomas, pero sus componentes están establecidos principalmente en inglés. | Para la internacionalización, angular cuenta con una librería (ngx-translate) que permite la carga dinámica de archivos de traducciones en nuestras aplicaciones angular. |
| Manejo de errores | Bootstrap al ser un framework de Front-end no cuenta con un manejador de errores. | Angular cuenta con diferentes implementaciones para el manejo de errores del lado del servidor, del lado del cliente y del servicio como tal. Algunas de estas son ErrorHandler, stacktrace.js, httpInterceptor, entre otras. |
| Soporte a nuevos  estándares | Bootstrap cuenta cuenta con actualizaciones para mantener la compatibilidad con html5, php. | El equipo de angular viene trabajando en hacer de este framework, un framework más estable, más escalable. |
| Mecanismos de  Licenciamiento | Se encuentra publicado bajo la licencia de “Apache 2 License” y todas sus versiones son open source. | Angular cuenta con la licencia MIT que es una licencia de software libre permisiva, compatible con DFSG (directrices de software libre de Debian) y GPL (General Public License ), aprobado por la FSF (free software foundation) y la OSI (open source initiative) |

2.3. Valoración de las características.

|  |  |  |
| --- | --- | --- |
|  | **Bootstrap** | **Angular** |
| Documentación | 9 | 9 |
| Seguridad | 5 | 8 |
| Facilidad de uso | 9 | 7 |
| Internacionalización | 6 | 9 |
| Manejo de errores | 0 | 8 |
| Soporte a nuevos  estándares | 9 | 9 |
| Mecanismos de  Licenciamiento | 9 | 9 |

2.4. Calificación de cada framework.

|  |  |
| --- | --- |
|  | Total |
| Bootstrap | 6.7 |
| Angular | 8.4 |

2.5. Cómputo de totales y selección del mejor framework.

Los criterios con los cuales realizamos esta evaluación dieron como candidato ganador a Angular, ya que cumplia con más características y mayor puntaje que Bootstrap, a pesar de esto consideramos que utilizar Angular para este proyecto no era necesario y que los elementos ofrecidos por Bootstrap son más fácil, y más rápidos de utilizar, además de que ya contábamos con experiencias anteriores con el mismo, por lo cual a pesar de que Angular contó con un mayor puntaje utilizamos Bootstrap el cual nos proporcionaba todo lo necesario para la realización del proyecto.

3. **Proceso de desarrollo:**

La realización estuvo dividida en varios momentos, lo primero fue plantear algunas de las tecnologías a implementar, para enumeramos algunas de las utilizadas:

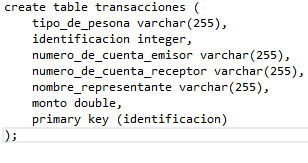
* Visual studio
* Realizado en .net
* MySql server 8.0.20
* UiPath
* Internet Information Services o IIS incluido en el visual studio y servidor web
* Mysql Workbrench

Se debe tener el cuenta que para que el bot rpa depure en algún navegador se debe descargar una extensión de uipath para el que se vaya a usar.

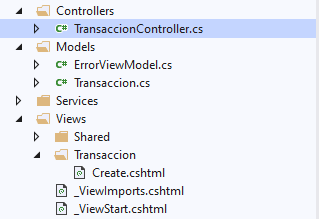
Descripción del backend:

A Continuación el script de la creación de la base de datos y la tabla respectiva al problema de la transacción que se nos plantea:

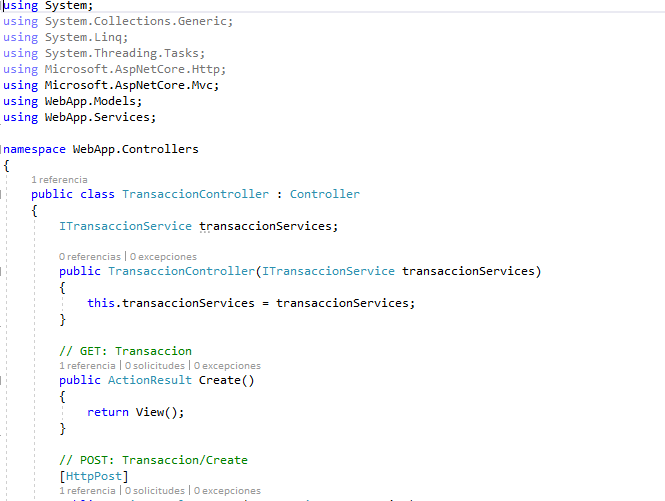


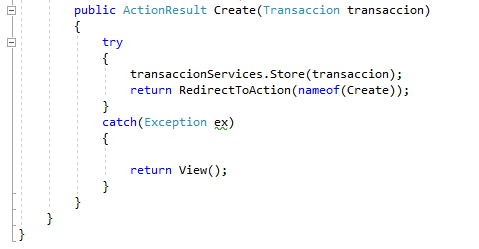


Después de haber creado la base de datos procedemos a realizar el el modelo y el controlador y se enlaza con la vista, obsérvese que el código está construido usando el modelo vista controlador el cual podemos ver en la distribución de sus paquetes:

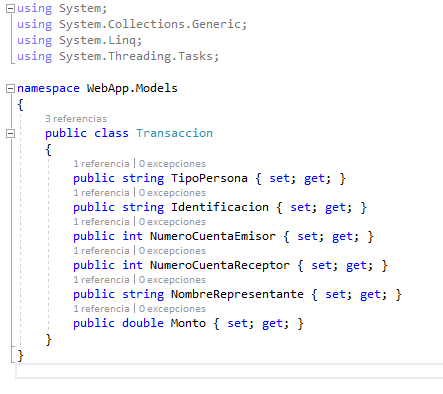


Veamos el código respectivo de cada capa:

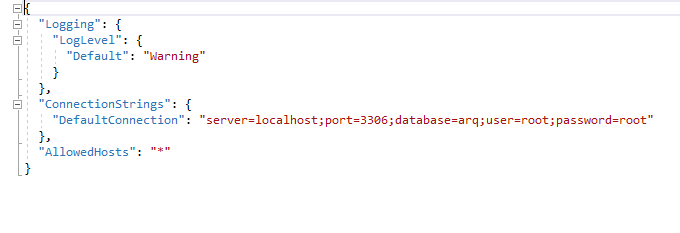




En el modelo podemos observar la declaración de cada campo requerido para la transacción y sus respectivos getter y setters:

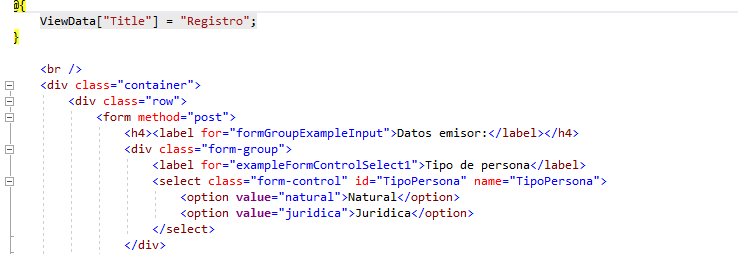


Para conectar la base de datos haremos uso de el archivo appsetting.json en cual asignamos la ruta correspondiente a nuestra aplicación web :



Y finalmente se procede con la creación del frontend el cual hace uso de bootstrap para ser un poco más amigable con el usuario y lograr que la app sea responsive:

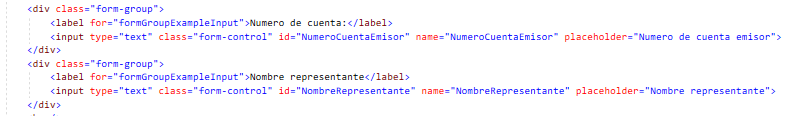
Primero creamos un selector del tipo de persona que realiza la transacción:



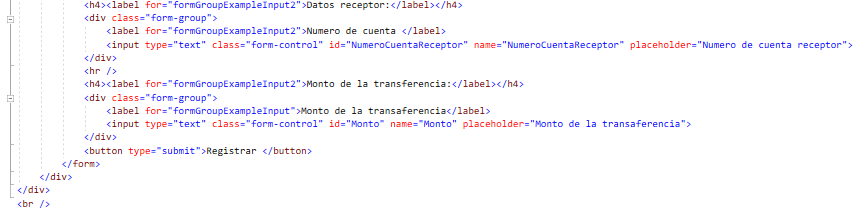
Luego el campo de entrada de texto que capturará el numero de identificacion (sea rut o nit ) de quien realiza la transacción :



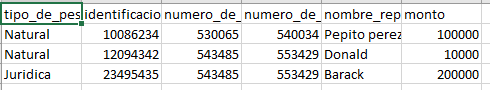
Continuamos con el número de cuenta que realizará la transacción y su respectivo nombre de representante:



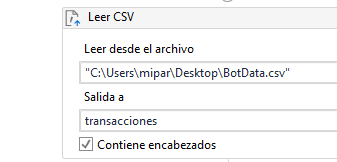
De la misma forma construimos los campos para la persona que recibirá el monto de la transacción y se pide el monto de dinero que se está efectuando en la transacción:



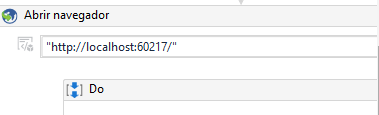
Finalizado el frontend procedemos a la implementación del bot rpa el cual lo hacemos a través de la plataforma de UiPath studio y usando el archivo .csv que para fines de prueba llenaremos con los siguiente tres campos:



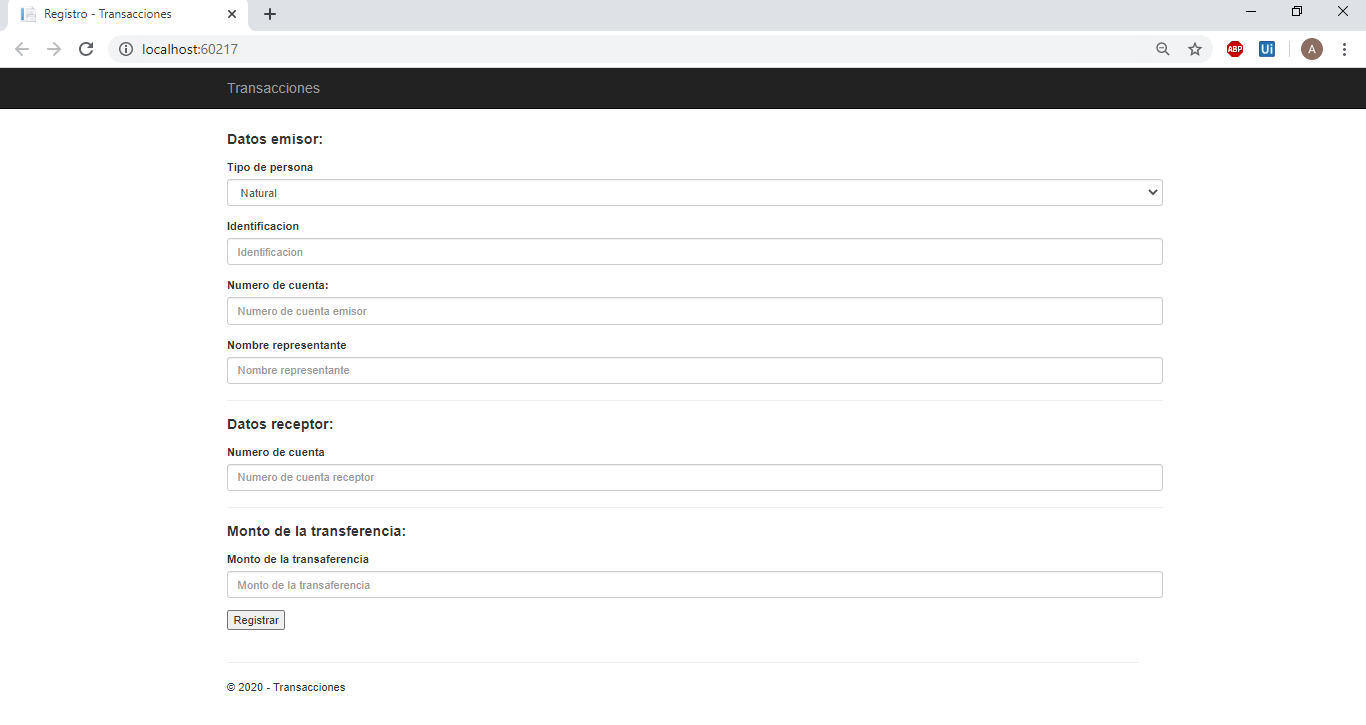
A este le tendremos que asignar la ruta de el archivo que contenga los datos que el bot debe ingresar en la base de datos:



Adicionalmente le damos la ruta de la aplicación web:



El resultado de la vista es el siguiente:



**BiBLIOGRAFÍA :**

**LINK REPOSITORIO GIT:**

**LINK VIDEO TEST BOT RPA:**