23-8-2022

Reto Técnico

Api para Convertir Divisa

Vanessa Burbano Quintero

BBVA COntinental

# INTRODUCCIÓN

El presente documento provee una visión general de la arquitectura del Software CONDIV, usando diferentes vistas para apreciar los diferentes aspectos del Software, las cuales están basados en los estándares de Interfaz de Programación de Aplicaciones (API), Transferencia de Estado Representacional (REST) y utilizando el Lenguaje de Modelamiento Unificado (UML).

# DEFINICIONES Y ABREVIACIONES

* API: Interfaz de Programación de Aplicaciones
* REST: Transferencia de Estado Representacional
* UML: Lenguaje de Modelamiento Unificado
* TOKEN: Permiten a un usuario autenticarse con las aplicaciones de Cloud y saltarse la verificación en dos pasos y el inicio de sesión único

# REFERENCIAS

* Procesos de Ingeniería de la Metodología Integrada de Osinergmin.
* IBM Corp. "Rational Unified Process® Versión 1.2" Rational IBM 2005.
* RED HAT “El concepto de las interfaces de programación de aplicaciones” 2017

# ALCANCE

* Realizar un tipo de cambio a un monto donde se deben utilizar el monto, moneda de origen, moneda de destino, monto con el tipo de campo y el tipo de cambio.
* Debe permitir el registro, actualización y búsqueda del tipo de cambio.
* Debe tener una autenticación.
* Por cada tipo de cambio realizado, se debe registrar quien hizo la solicitud (auditoría funcional).

# METAS Y RESTRICCIONES DE LA ARQUITECTURA

El software CONDIV pretende ofrecer la facilidad al usuario final de convertir dinero en distintas monedas; este software se desarrolla con .NET CORE y ADO.NET Entity Framework para generar una aplicación rápida y robusta que pueda soportar muchas transacciones por minuto.

Con la finalidad de otorgar un software seguro, es implementando JWT bajo la librería Microsoft AspNetCore Authentication JwtBearer, esto permite restringir el acceso a los módulos a usuarios que no tengan el permiso debido.

# DESCRIPCIÓN GENERAL DEL SISTEMA

## PROCESO

Diagrama

Descripción generada automáticamente

El Software estará constituido por los siguientes componentes:

* **TOKEN**

Este componente permite validar al usuario y en caso de que el usuario sea un usuario activo se le asignará un token que usará para realizar transacciones en los otros componentes del sistema

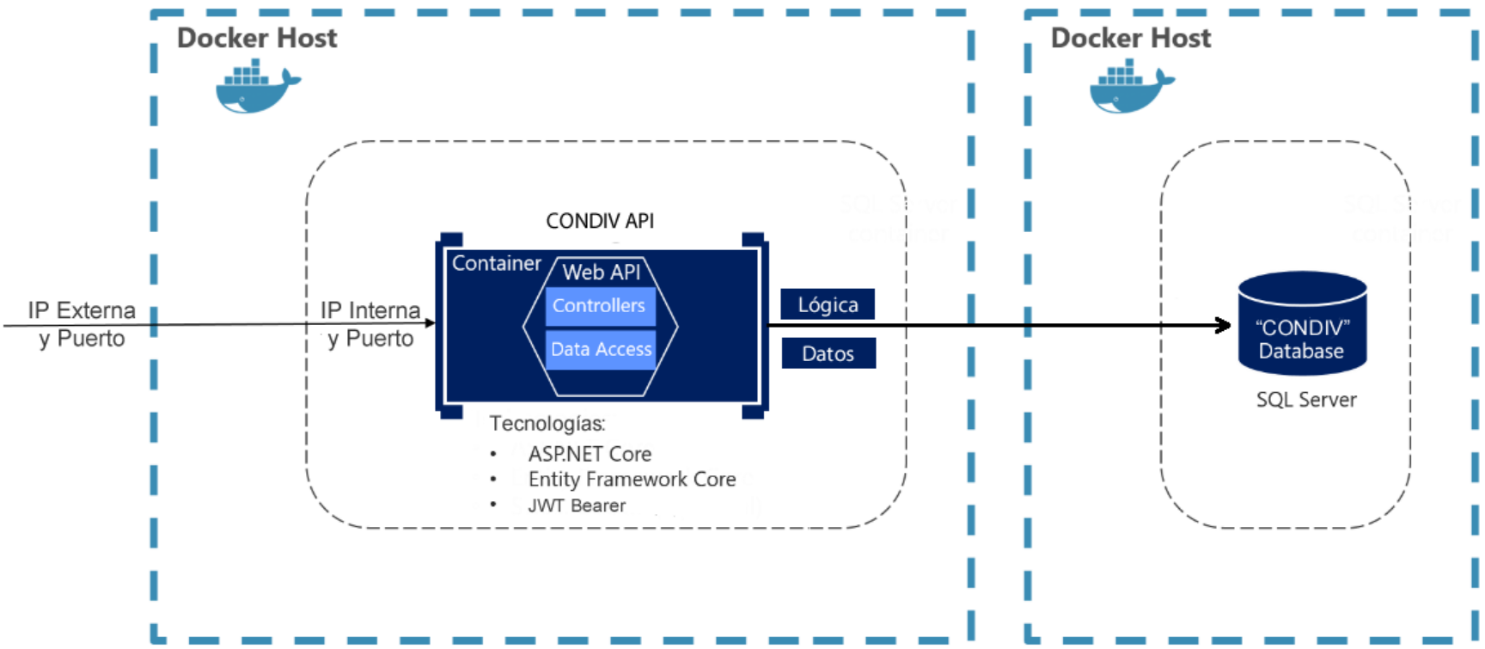
* **Convertir Moneda**

Este componente permite convertir un monto de dinero de una moneda a otra, la cual el usuario final usará para cotizar el tipo de cambio.

* **Registrar Tipo de Cambio**

Este componente registra el tipo de cambio actual y desactiva el tipo de cambio anterior, con un valor de factor de cambio para las transacciones (compra o venta).

# DEFINICIÓN DE LA ARQUITECTURA



## CAPAS DE LA ARQUITECTURA

La aplicación ha sido desarrollada con la tecnología .NET CORE de Microsoft, actual entorno de Microsoft para desarrollo de aplicaciones, con una arquitectura de N-capas. Estará compuesta por las siguientes capas:

* **Lógica de Negocios:** 
  + Capa de objetos de lógica de negocios: componentes encargados de ejecutar procesos complejos de la lógica de negocio, que interactuarán con la capa de Capa de acceso de datos.
  + Capa de lógica de acceso de datos: separa limpiamente la lógica de acceso a datos de la lógica.
* **Datos:** 
  + Capa de implementación de acceso a datos: ADO.NET Entity Framework.

Diagrama

Descripción generada automáticamente

## ARQUITECTURA DE INFRAESTRUCTURA

### PLATAFORMA DE SOFTWARE BASE

**Cliente:**

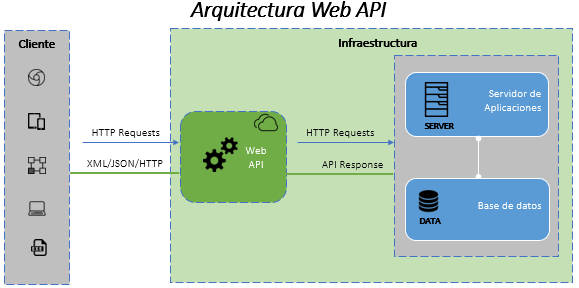
|  |  |
| --- | --- |
| **Sistema Operativo** | Multiplataforma |
| **Tecnologías utilizadas** | Toda tecnología que soporte consumo de webservices |
| **Lenguajes Utilizados** | Json |

**Servidor de Software:**

|  |  |
| --- | --- |
| **Sistema Operativo** | Multiplataforma |
| **Lenguajes Utilizados** | .NET CORE 6 |
| **Framworks Utilizados** | ADO.NET Entity Framework, Microsoft AspNetCore Authentication JwtBearer |

**Servidor de base de datos:**

|  |  |
| --- | --- |
| **Sistema Operativo** | Multiplataforma |
| **Base de Datos** | SQL Server 2017 |

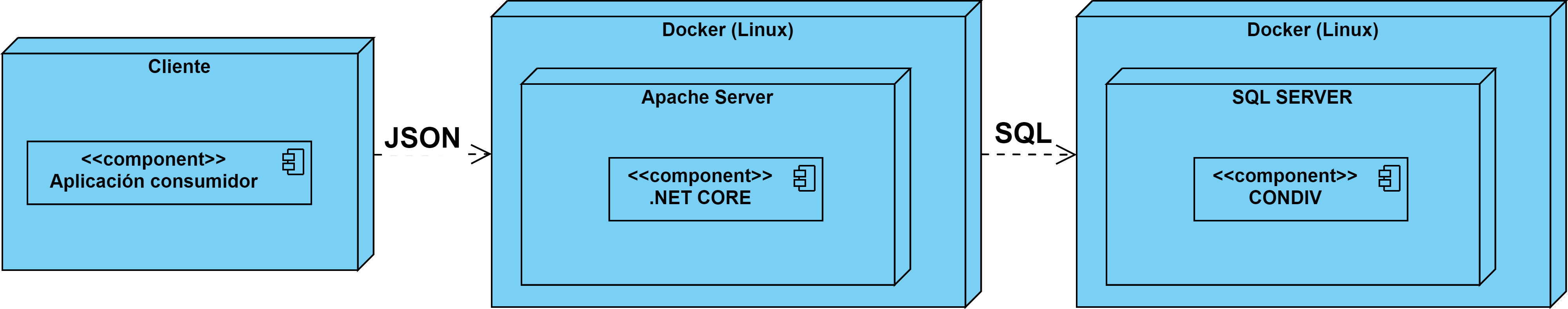
­­ 

# DIAGRAMA DE DESPLIEGUE

Pretendemos a través de este diagrama mostrar la arquitectura de ejecución del sistema CONDIV

El diagrama representa cómo y dónde se desplegará el sistema.

A continuación, se muestra el diagrama de Despliegue del Software:

**

Las máquinas físicas se representan como nodos.

El nodo es un elemento donde se ejecutan los componentes

Los componentes embebidos son aplicaciones, librerías y motor de base de datos.

Entre los nodos y componentes se comunican a través de relaciones lo que indica el tipo de conexión entre ellos.

El Software estará desplegado en dos servidores el primero en el cual se encontrará procesando el servidor de aplicaciones (Apache Server) y el otro se encontrará procesando el gestor de base de datos (SQL SERVER 2017).

# DIAGRAMA DE COMPONENTES

El diagrama de Componentes muestra los módulos en los que estará dividida la funcionalidad del sistema. Los conectores identifican la relación entre los mismos.

El Software cuenta con módulos claramente definidos los cuales acceden al modelo de datos único de la solución. Los módulos se dividen en: Módulo TOKEN, Convertir Moneda, Registrar Tipo de Cambio.

Interfaz de usuario gráfica, Diagrama

Descripción generada automáticamente

**.NET CORE:** Framework para generar y devolver los datos en formato json.

**Módulo TOKEN:** Componente que verifica y devuelve un hash de autenticación en caso la verificación sea exitosa.

**Módulo Convertir Moneda:** Componente que convierte el monto de una moneda a otra y registra un historial de cotización.

**Módulo Registrar Tipo de Cambio:** componente que registra el tipo de cambio del día.

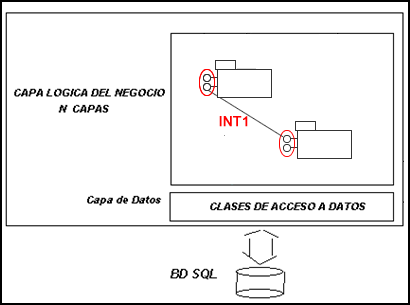
**ADO.NET Entity Framework:** Es una herramienta que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de un Software.

**SQL Server:** Motor de Base de Datos seleccionado para el Software.

# VISTA DE INTEGRACIÓN

## DIAGRAMA DE INTEGRACIÓN E INTERFACES

Para el desarrollo de este software utilizaremos 2 capas (Capa de objetos de lógica de negocios y Capa de lógica de acceso de datos) las cuales se comunicarán con la base de datos y retornarán la información solicitada



|  |  |  |
| --- | --- | --- |
| **Interfaz** | **Descripción Breve** | **Tipo de Interfaz** |
| INT1 | La interfaz 1 contiene las reglas de negocio, la cual se encarga de generar tokens, convertir dinero y actualizar el tipo de cambio, se apoya con la interfaz 2 la cual apoya creando la conexión a la base de datos | Interfaz Interna |

## CRITERIOS PARA EL DISEÑO Y SELECCIÓN DE INTERFACES

Para la selección y el diseño de las diferentes alternativas de las interfaces se evaluarán los siguientes criterios:

* Facilidad de Construcción
* Tiempo de Respuesta
* Consumo de recursos
* Seguridad

## CRITERIOS DE INTEGRACIÒN DEL SOFTWARE

Para la óptima integración del Software se deberán tener que cumplir, considerar y evaluar los siguientes criterios:

* Antes de realizar la integración todos los componentes deberán haber pasado por pruebas unitarias.
* Antes de realizar la integración, todas las incidencias, errores u otras no conformidades encontradas durante las pruebas unitarias deberán estar cerradas.
* Se deberá tener preparado los ambientes y entornos para la integración (Entorno de Desarrollo o Entorno de Integración).
* Deberá haberse inicializado y migrado data consistente previa a la integración.
* Otros Criterios que apoyen a que la integración resulte un éxito.

## SECUENCIA DE INTEGRACIÒN

Para que el Software se integre totalmente se seguirá la siguiente secuencia de integración:

* Realizar las pruebas unitarias a todos los componentes desarrollados (De todos los módulos).
* Levantar todos los errores e incidencias encontradas en las pruebas unitarias (De todos los módulos).
* Realizar revisión de pares al código fuente y levantar las no conformidades.
* Asegurarse que todos los componentes del Software estén completamente corregidos (Realización de nuevas pruebas sobre los errores encontrados).
* Validar que el entorno de integración esté listo.
* Validar que la data haya sido migrada satisfactoriamente.
* Iniciar la integración
  + Integrar Modulo TOKEN - Realizar pruebas de integración.
  + Integrar Modulo TOKEN y Modulo Convertir Moneda - Realizar pruebas de integración entre módulos.
  + Integrar Modulo TOKEN y Modulo Convertir Moneda y Modulo Registrar Tipo de Cambio - Realizar pruebas de integración entre módulos.
* Finalmente realizar las pruebas del Software y luego de ellas las Pruebas de Aceptación con los Usuarios Finales.

# CONTROL DE VERSIONES

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión N°** | **fecha** | **Autor** | **Descripción** |
| 1.0 | 23/08/2022 | Vanessa Burbano | Creación del Documento |
|  |  |  |  |