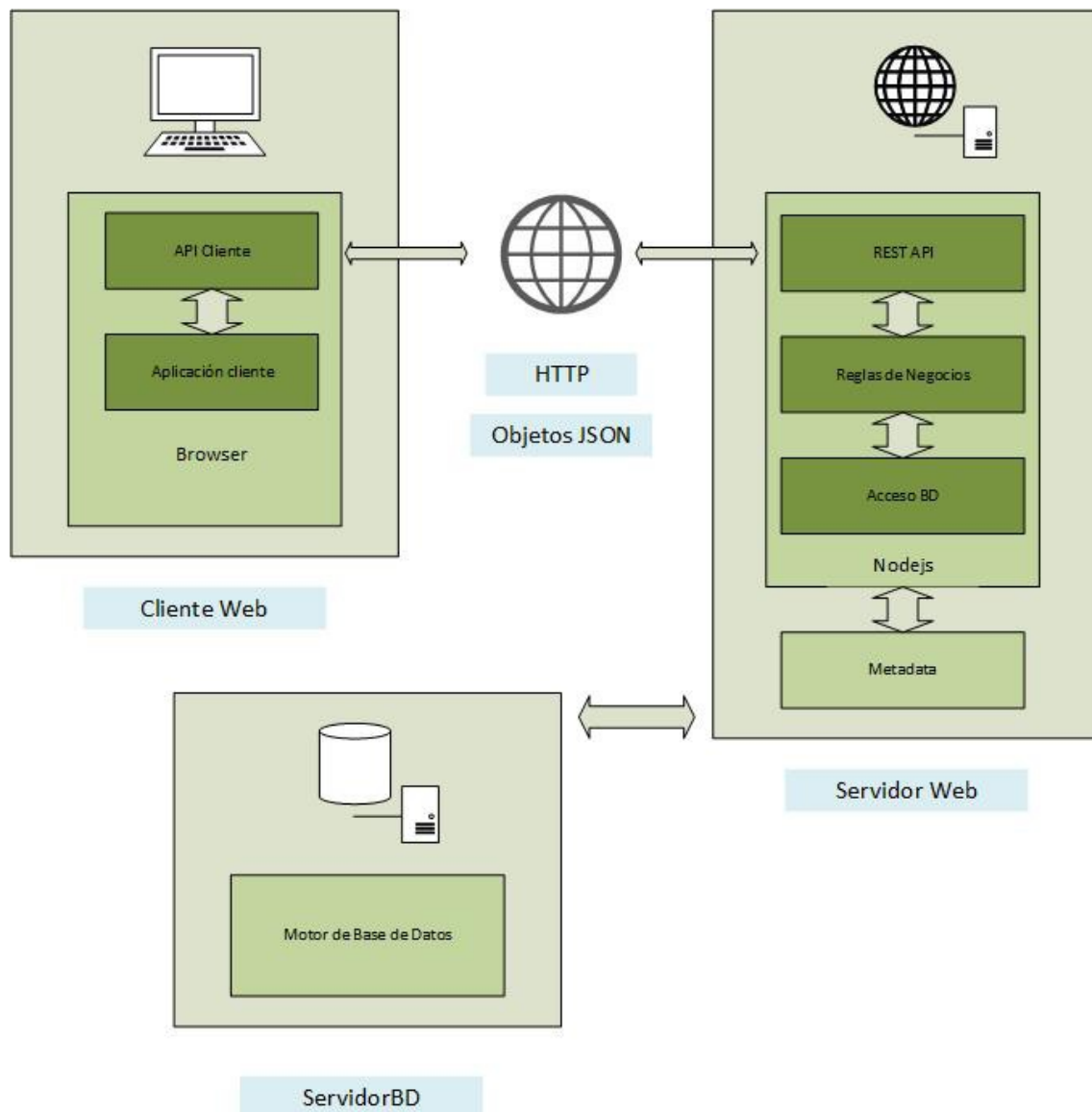


Nodejs REST API

Arquitectura

Este documento es una presentación de la arquitectura de la REST API.

La implementación de la REST API responderá a una arquitectura de diseño en tres capas expresada en el siguiente diagrama:



El diseño en tres capas es conceptual, lo que significa que en el modelo real podrá implementarse en dos o una capas.

Servidor Web.

El servidor web se implementa sobre Nodejs con un middleware de router Express.

En Express se definirán los “end points” de la API, que en principio serán siete.

La programación del server Nodejs toma datos del modelo de la Base de Datos estructurados como “Recursos” a los cuales se les aplicarán los cuatro verbos REST.

Estos datos están almacenados como “metadata” (archivos JSON) que proveen la información necesaria para procesar la validación de la información recibida por la API, previo a su impacto en la Base de Datos, evitando tanto “round trips” a la base como interpretación de errores SQL.

La API genera en forma dinámica tanto las consultas a la base como las actualizaciones a la misma.

La API maneja las transacciones contra la base y provee control de concurrencia a nivel fila de las tablas de la base (recursos).

La programación del Nodejs es asincrónica con manejo de “promises” y secuenciado con “async/await”.

Las reglas de negocios se deberán programar de acuerdo a este paradigma, devolviendo “promises”.

Estas reglas (o su invocación) deberá insertarse en el código provisto para que tenga la capacidad de ser ejecutado en las distintas etapas del “ciclo de procesamiento”.

El servidor Web expondrá una API en formato JSON para los verbos HTTP que soporten “body” (POST, PUT, DELETE) y un intérprete del “query string” para el verbo GET.

Cliente Web.

El cliente web (fuera del alcance de este proyecto) deberá contemplar el procesamiento asincrónico del consumo de la API interpretando los códigos de retorno y el formato JSON del “response body”.

En el próximo envío de detallarán los “endpoints” y su funcionalidad, el formato de las interfaces y la estructura de la Metadata.