

Unidad 2 – parte 2

Formatos de intercambio

Bases de Datos Aplicada

v1.0 – Abril 2024



Universidad Nacional
de La Matanza

DIIT
Departamento de Ingeniería e
Investigaciones Tecnológicas

Contenido

- Formatos de intercambio
- Texto simple, separado por comas, XML, JSON.
- APIs.

Al finalizar deberías ser capaz de...

- Identificar e interpretar distintos formatos de intercambio.
- Incorporar a una DDBB datos procedentes de XML, JSON, CSV, etc.
- Comprender qué es una API web y cómo interactuar a un nivel básico con ella.

Formatos de intercambio de datos

Estándares ampliamente difundidos para almacenar y compartir información entre sistemas.

Adherir a un estándar implica que no necesitamos saber de antemano quién utilizará los datos. Reduce además la información adicional requerida para interpretarlo.

Siempre de texto simple.

Poseen estructuras y reglas (sintaxis) tal como los lenguajes de programación.

Suelen existir funciones de biblioteca para generarlos o interpretarlos.

Ejemplos: XML, JSON, YAML, CSV, EDI, etc.

EDI (Electronic Data Interchange)

Formato de texto simple (ASCII, Unicode, etc).

Orientado a funcionar computer-to-computer.

Existen varios estándares en uso, cada uno con múltiples versiones.

Debe informarse el estándar adherido en el intercambio.

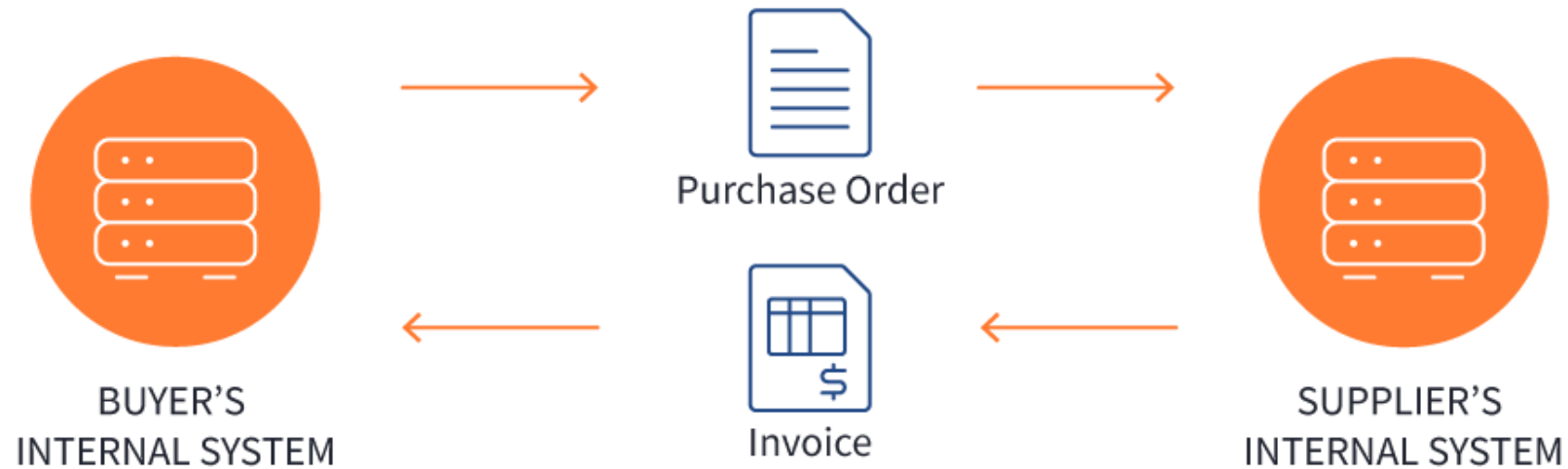
Se detalla cómo interpretar números, fechas, etc.

Ejemplos: ANSI, ASC, X12, EANCOM, HIPAA, ODETTE, SWIFT, etc.

Usos comunes: facturas, órdenes de compra.

Más información en: <https://www.edibasics.com/what-is-edi/>

EDI: ejemplo



Fuente: <https://www.edibasics.com/what-is-edi/>

EDI: ejemplo

ST*810*0001~BIG*20000513*SG427254*20000506*508517*1001~N1*ST*ABC AEROSPACE CORPORATION*9*123456789-0101~N3*1000 BOARDWALK DRIVE~N4*SOMEWHERE*CA*98898~ITD*05*3*****30*****E~IT1*1*48*EA*3**MG*R5656-2~TDS*14400~CTT*1~SE*10*0001~

Tabla 1 (de 3).

ST*810*0001~

Indica inicio de una factura y asigna un número de control.

BIG*20000513*SG427254*20000506*508517*1001~

La factura fue emitida el 13/05/2000 y se le asignó el número SG427254. Corresponde a la orden de compra número 508517, envoi 1001 enviada el 06/05/2000.

N1*ST*ABC AEROSPACE CORPORATION*9*123456789-0101~

El domicilio de entrega es “ABC Aerospace Corporation” y un identificador fiscal.

N3*1000 BOARDWALK DRIVE~

Domicilio de entrega Boardwalk drive 1000.

Fuente: <https://x12.org/examples/004010x348/example-01-basic-invoice>

CSV: Comma separated values

Formato de texto simple (ASCII, Unicode, etc).

Interpretación tabular separando registros por filas.

Las columnas se separan con un carácter delimitador (en gral coma o punto y coma).

También existe el TSV: separado por carácter tabulador.

Cada registro (fila) debe tener la misma cantidad de campos.

Si no hay valor para un campo se presentan dos delimitadores contiguos.

Pueden contener nombres de campo en la primera fila.

El delimitador se debe escapar o indicar entrecomillado si es parte de un dato.

CSV: ejemplo



GOBIERNO DE LA PROVINCIA DE
BUENOS AIRES

Dataset

Datos Abiertos PBA / Dataset

Nacimientos

Ministerio de Gobierno. Subsecretaría de Gestión Operativa. Dirección Provincial del Registro de las Personas.

Nacimientos inscriptos en el Registro Provincial de las Personas.

Recursos del dataset



Registro mensual de nacimientos

Nacimientos registrados mensualmente por el Registro Provincial de las Personas con información de municipio y delegación. Periodo: 2016-202...

CONSULTAR

DESCARGAR

CSV: ejemplo

```
anio,mes,zona,municipio_id,municipio_nombre,delegacion,genero,cantidad
2019,1,17,06427,La Matanza,San Justo,masculino,74
2019,2,17,06427,La Matanza,San Justo,masculino,73
2019,3,17,06427,La Matanza,San Justo,masculino,66
2019,4,17,06427,La Matanza,San Justo,masculino,59
2019,5,17,06427,La Matanza,San Justo,masculino,91
2019,6,17,06427,La Matanza,San Justo,masculino,49
2019,7,17,06427,La Matanza,San Justo,masculino,69
2019,8,17,06427,La Matanza,San Justo,masculino,58
2019,9,17,06427,La Matanza,San Justo,masculino,52
2019,10,17,06427,La Matanza,San Justo,masculino,50
2019,11,17,06427,La Matanza,San Justo,masculino,41
2019,12,17,06427,La Matanza,San Justo,masculino,44
```

Fuente: <https://catalogo.datos.gba.gob.ar/dataset/nacimientos>

Ancho fijo, etc.

Formato de texto simple (ASCII, Unicode, etc).

Existen variantes con una premisa similar.

Ancho fijo

- Cada registro en una fila.
- Cada columna tiene una cantidad determinada de caracteres.
- Debe determinarse carácter de relleno (típicamente ceros en números y espacios en texto).
- Cada registro tiene la misma cantidad de caracteres.

Tabulados

- Símil CSV pero utilizando el carácter tabulador.

XML: eXtended Markup Language

Formato de texto simple (ASCII, Unicode, etc).

Similar a HTML (evolución de este).

Autodescriptivo. Cada dato se delimita por etiquetas.

```
<etiqueta>dato</etiqueta>
```

No hay etiquetas predefinidas ni una estructura de documento.


Se puede usar sangría (indentación) para facilitar la interpretación humana.

Admite el anidado de tablas.

Ampliamente difundido para aplicaciones de formatos de archivo.


Admite la generación de un archivo de plantilla

XML: ejemplo

Popular Baby Names  189 recent views

City of New York — Popular Baby Names by Sex and Ethnic Group Data were collected through civil birth registration. Each record represents the ranking of a baby name in the order of...

[CSV](#) [RDF](#) [JSON](#) [XML](#)



Fuente: https://catalog.data.gov/dataset/?q=baby+names&res_format=XML

XML: ejemplo

```
<response>
  <row _id="row-v4f5~xz3v-vr86">
    <brth_yr>2011</brth_yr>
    <gndr>FEMALE</gndr>
    <ethcty>HISPANIC</ethcty>
    <nm>GERALDINE</nm>
    <cnt>13</cnt>
    <rnk>75</rnk>
  </row>
  <row _id="row-gdep~mr7x-dj3u">
    <brth_yr>2011</brth_yr>
    <gndr>FEMALE</gndr>
    <ethcty>HISPANIC</ethcty>
    <nm>GIA</nm>
    <cnt>21</cnt>
    <rnk>67</rnk>
  </row>
```

Fuente: https://catalog.data.gov/dataset/?q=baby+names&res_format=XML

JSON: JavaScript Object Notation

Formato de texto simple (ASCII, Unicode, etc).

Utiliza una estructura jerárquica y admite valores anidados.

Emplea llaves { } para almacenar objetos y corchetes [] para almacenar vectores.

- Utiliza la lógica de pares clave/valor (key/value).
- Los objetos consisten en uno o más pares de claves/valores (key/value) contenidos dentro de llaves { }.
- Las claves deben ser cadenas de caracteres contenidas dentro de comillas " ".
- Los valores deben ser un tipo de información válida (cadena de caracteres, números, vectores, valores booleanos, caracteres nulos u otro objeto).

JSON: JavaScript Object Notation

Las claves y los valores son separados por dos puntos (:).

Múltiples pares de claves /valores dentro de un objeto se separan mediante comas.

El espacio en blanco no es significativo.

Para representar vectores (de variables escalares, objetos, etc):

- La clave debe estar seguida por dos puntos (:) y una lista de valores contenidos dentro de corchetes [].
- Un vector en JSON es una lista de un tipo de valor de entre los existentes (caracteres, números, valores booleanos, objetos u otro vector).

Cada valor dentro del vector es separado por una coma.

JSON: Ejemplo

The screenshot shows the 'datos.gob.es' website. The header includes the Spanish government logo and the text 'datos.gob.es reutiliza la información pública'. The main navigation bar has links for 'APORTA INITIATIVE', 'DATA CATALOG', 'IMPACT', 'SECTORS', 'CURRENT NEWS', 'INNOVATION', and 'INTERACT'. Below this is a secondary bar with 'DATASETS', 'API', and 'SPARQL ENDPOINT'. The breadcrumb trail reads 'Home | Data Catalog | Datasets | Obras más consultadas en la ...'. The dataset title is 'Obras más consultadas en la Biblioteca Digital Hispánica en 2019'. The category is 'Culture and leisure'. The publisher is 'Biblioteca Nacional de España (MINISTERIO DE CULTURA Y DEPORTE)'. The administration level is 'State Administration'. The license is 'https://creativecommons.org/publicdomain/zero/1.0/deed.es_ES'. The description states: 'Listado con las diez obras más consultadas en la Biblioteca Digital Hispánica en 2019. Cómo citar el conjunto: Título del set de datos. [Conjunto de datos]. Versión de DDMMAAAA. Datos.gob.es. URL del dataset Ej. Archivo de la Web Española: Comunidad Autónoma de Aragón. [Conjunto de datos]. Versión de enero de 2019. Datos.gob.es. https://datos.gob.es/es/catalogo/ea0019768-archivo-de-la-web-espanola-comunidad-autonoma-de-aragon'. The contact is 'bnelab@bne.es'.

Fuente: <https://datos.gob.es/en/catalogo/ea0019768-obras-mas-consultadas-en-la-biblioteca-digital-hispanica-en-2019>

JSON: ejemplo

```
[
  {
    "Posición": "1",
    "Obra": "Poema del Cid.",
    "Enlace_BDH": "http://bdh.bne.es/bnearch/detalle/bdh0000036451"
  },
  {
    "Posición": "2",
    "Obra": "Beato de Liébana: código de Fernando I y Dña. Sancha.",
    "Enlace_BDH": "http://bdh.bne.es/bnearch/detalle/bdh0000051522"
  },
  {
    "Posición": "3",
    "Obra": "Índices de varias obras genealógicas, S. XVIII [Manuscrito]. Signatura: MSS/11490.",
    "Enlace_BDH": "http://bdh.bne.es/bnearch/detalle/bdh0000192085"
  }
]
```

Fuente: www.bne.es/sites/default/files/redBNE/datosgob/estadisticas/bdh/masconsultadas/consultadas_2019.json

YAML: YAML Ain't a Markup Language

- Se considera una versión mejorada de JSON.
- Minimalista, más fácil de leer y escribir.
- Utiliza la sangría para definir la estructura (como Python).
 - Pero no admite tabuladores. Siempre espacios (uno o más).
- Objeto YAML se compone de uno o más pares clave-valor.
- El par clave-valor se separa por dos puntos, sin comillas (excepto cadenas).
- Se utiliza un guión para separar elementos en una lista.
- Un documento (un archivo puede contener varios) comienza con tres guiones.

YAML: Ejemplo

```
%YAML 1.2
```

```
---
```

```
YAML: YAML Ain't Markup Language™
```

What It Is:

YAML is a human-friendly data serialization language for all programming languages.

YAML Resources:

YAML Specifications:

- YAML 1.2:
 - Revision 1.2.2 # Oct 1, 2021 *New*
 - Revision 1.2.1 # Oct 1, 2009
 - Revision 1.2.0 # Jul 21, 2009
- YAML 1.1
- YAML 1.0

Fuente: <https://yaml.org>

YAML: Ejemplo

```
# mongod.conf

# Where and how to store data.
storage:
  dbPath: C:\Program Files\MongoDB\Server\4.4\data
  journal:
    enabled: true
# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: C:\Program Files\MongoDB\Server\4.4\log\mongod.log

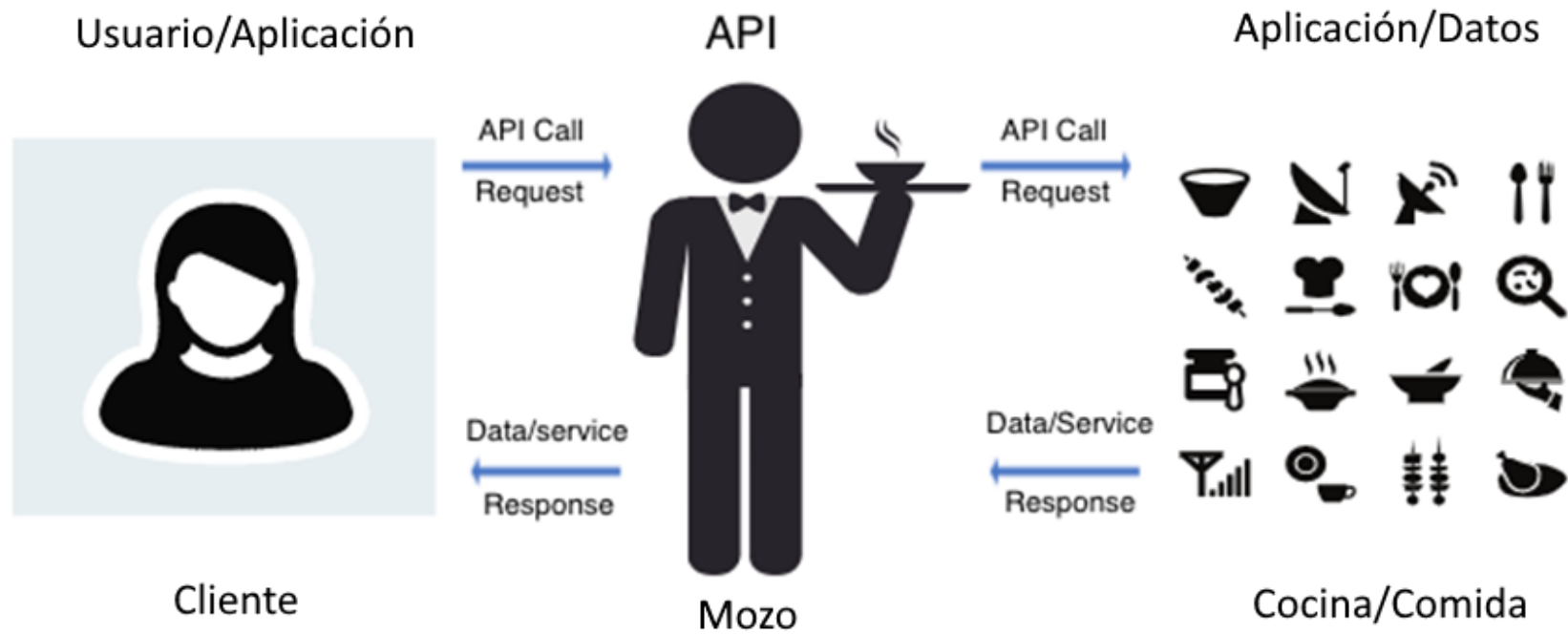
# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1
```

Práctica sugerida

- Conversión entre formatos (excepto EDI).
 - Por ejemplo: <https://www.json2yaml.com/>
- Operaciones de inserción en SQL por lote (bulk).
- Consulta directa desde SQL de campos XML, JSON.
- Generación de XML, JSON desde SQL.
- Utilización de datasets públicos:
 - <https://catalog.data.gov/>
 - <https://catalogo.datos.gba.gob.ar/>
 - <https://datos.gob.es/>

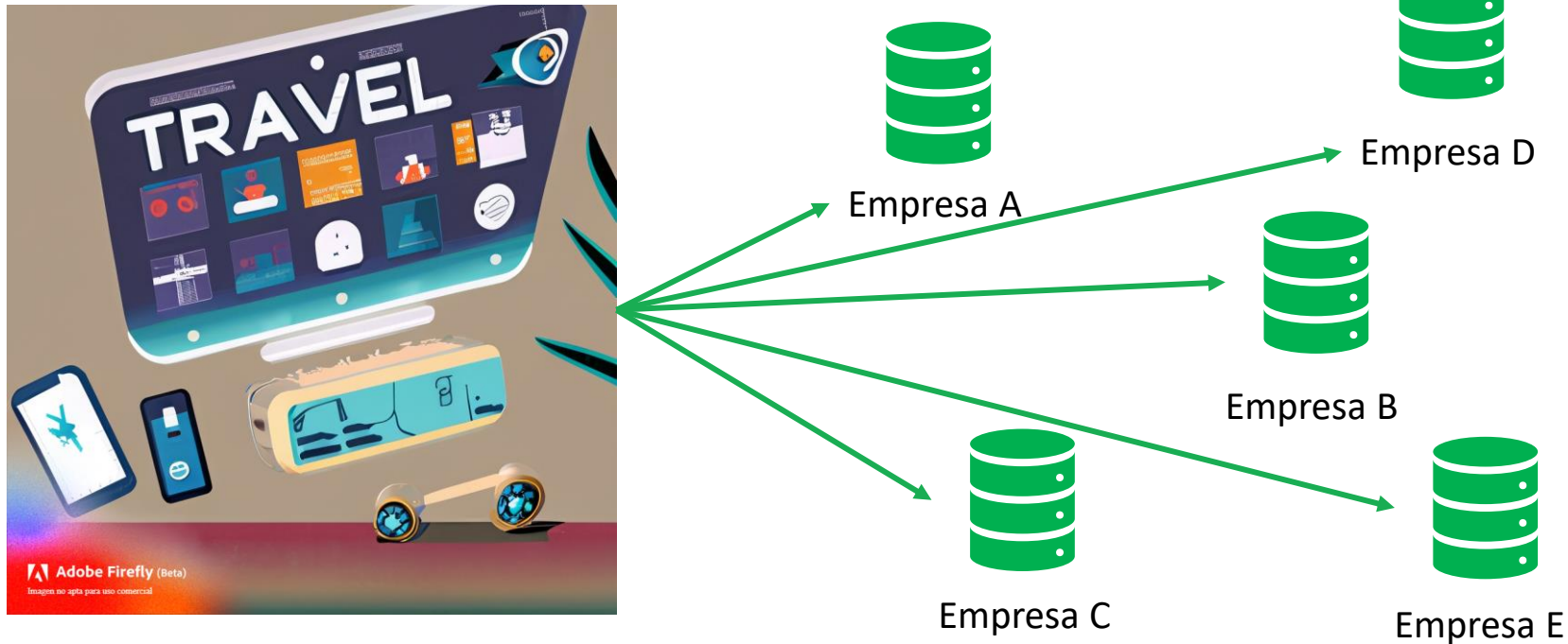
API: Application Programming Interface

Contrato o acuerdo implementado en software que permite el acceso a datos o servicios de un sistema informático desde otro sistema informático.



APIs: Ejemplo

Un sistema de una agencia de turismo se comunica con distintos sistemas de empresas de transporte terrestre, aéreo, marítimo, y permite reservar y comprar pasajes.



APIs

APIs abiertas o públicas: disponibles para su uso gral. Algunas requieren una clave (o un token) para controlar las solicitudes que atiende por cliente. También pueden limitar las peticiones según el plan contratado. Ejemplo: Google Maps.

APIs internas o privadas: se utilizan dentro de una organización.

APIs de socios: permiten que una organización intercambie información con proveedores, clientes o socios comerciales.

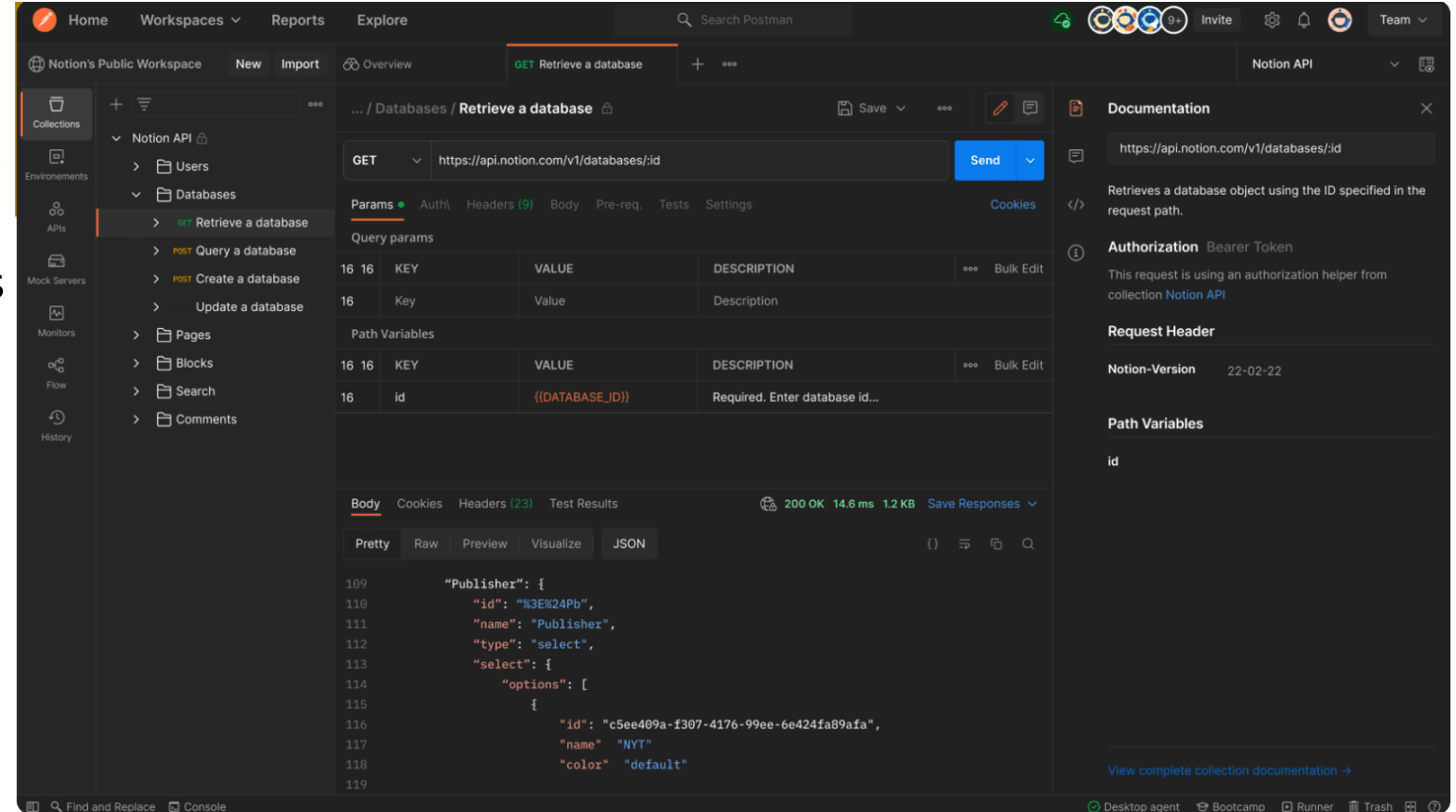
Tipos de comunicación: SOAP, XML-RPC, JSON-RPC, REST.

Las API REST o API HTTP son las más difundidas.

APIs: Postman

Postman es una plataforma de APIs para crear y usar APIs. Simplifica cada paso del ciclo de vida de las API y facilita la colaboración para crear mejores APIs y hacerlo más rápido.

Recursos adicionales en:
Postman Academy



Fuente: <https://www.postman.com/>

¿Dudas?



Universidad Nacional
de La Matanza

DIIT



Departamento de Ingeniería e
Investigaciones Tecnológicas