PowerBI Embedded

Funcionamiento

BIG DATA – BUSINESS INTELLIGENCE – MACHINE LEARNING

Contenido

1. INTRODUCCIÓN 3
2. REQUISITOS 3
3. EMBEDDED PARA CLIENTES 3
   1. ESQUEMA DE USO PBIE PARA CLIENTES 3
   2. PASO A PRODUCCIÓN 4
   3. CÓDIGO 7
4. EMBEDDED PARA ORGANIZACIÓN 9
   1. ESQUEMA DE USO PBIE PARA ORGANIZACIÓN 9
   2. PASO A PRODUCCIÓN 9
   3. CÓDIGO 9
5. FUNCIONAMIENTO ARQUITECTURA 12
   1. FLUJO DE INFORMACIÓN 12
   2. PETICIÓN DEL CLIENTE 13
   3. SERVICIO EXPRESS 13
6. TECNOLOGÍAS 15
7. INFORMACIÓN SOBRE STRATEBI 17
8. OTROS 18
9. EJEMPLOS DE DESARROLLOS ANALYTICS 19

1. INTRODUCCIÓN

El servicio PowerBI (SaaS) y el servicio PowerBI Embedded en Azure (PaaS) tienen las API para insertar los paneles e informes…

PowerBI Embedded está concebido para simplificar la manera en que los ISV y los desarrolladores usan las funcionalidades de PowerBI con análisis integrado. PowerBI Embedded simplifica las funcionalidades de Power BI al ayudarle a agregar rápidamente objetos visuales, informes y paneles a las aplicaciones.

PowerBI Embedded está destinado a fabricantes de software independientes (ISV) y a desarrolladores que compilan aplicaciones para sus clientes. Se puede usar como un servicio de inteligencia empresarial de terceros que permite visualizar datos de la aplicación.

PowerBI Embedded es una característica de pago por uso.

2. REQUISITOS

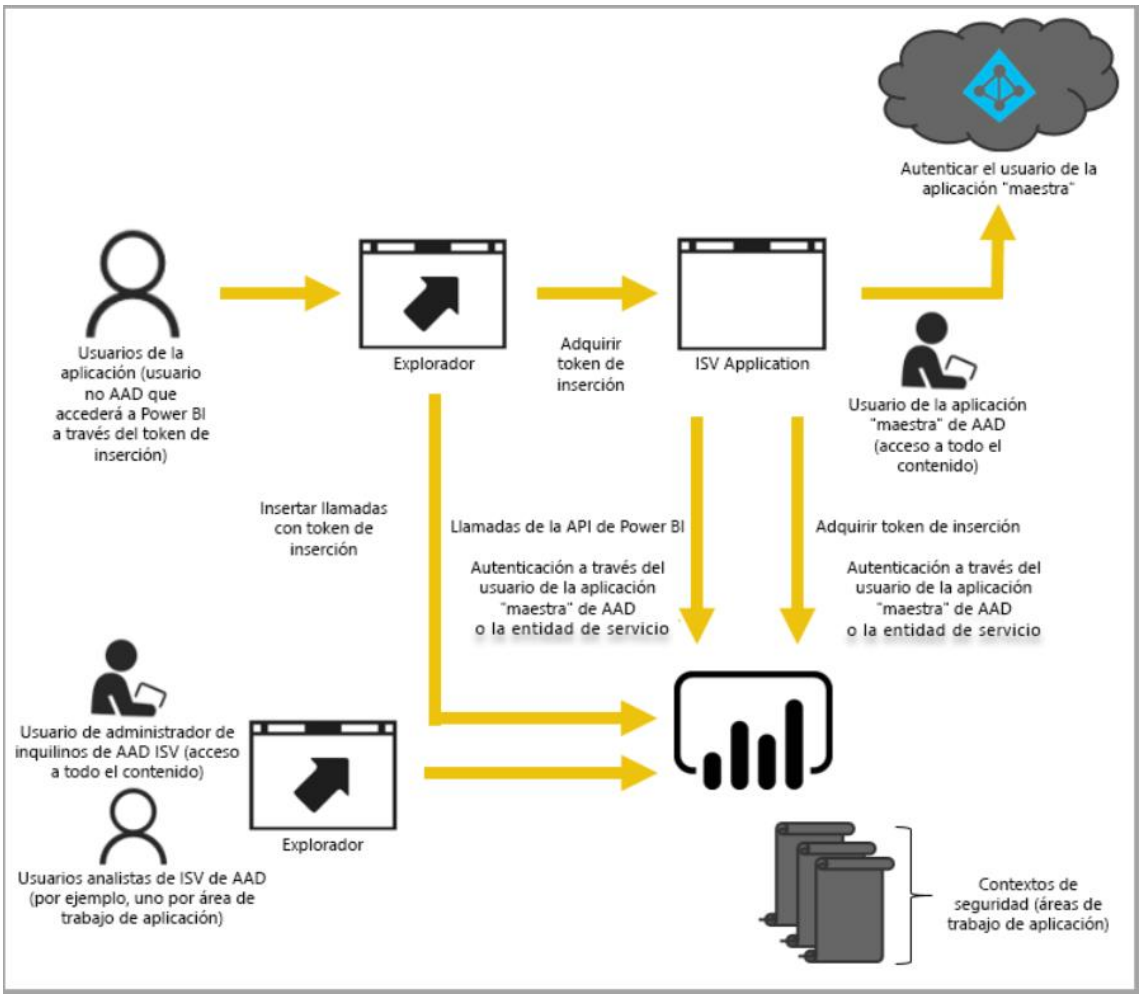
Distinguimos dos posibilidades de embeber contenido de PowerBI en una aplicación o servicio web, embedded para clientes y embedded para organización.

- Embedded para clientes:
  - Una cuenta PRO (para desarrollo).
  - Un tenant de Azure Active Directory.
  - Instancias de Azure con PowerBI Embedded o capacidad Premium.
  - PowerBI Desktop.
- Embedded para organización:
  - Cuentas PRO para los usuarios de la organización que tienen acceso a la aplicación web.
  - Capacidad Premium (OPCIONAL, sólo si hay usuarios externos a la organización).
  - Un tenant de Azure Active Directory.
  - PowerBI Desktop.

3. EMBEDDED PARA CLIENTES

La inserción de contenido para los clientes permite insertar paneles e informes a los usuarios que no tienen una cuenta de PowerBI.

1. Esquema de uso PBIE para clientes



En el caso de desarrollo tenemos una cuenta PRO que actúa como proxy, sería la cuenta maestra que autentica de manera automática a cualquier usuario que quiera ver los objetos de PowerBI embebidos en la aplicación.

2. Paso a Producción

Como hemos comentado previamente necesitaremos una capacidad contratada para el paso a producción, a continuación, se explica qué capacidades ofrece Microsoft y las tarifas correspondientes a cada una.

- PowerBI Premium: Es una suscripción de Office 365 de nivel de inquilino disponible en dos familias de SKU, EM y P.
- PowerBI Embedded: Es una capacidad dedicada desde Microsoft Azure Portal. Esta suscripción utiliza las SKU de tipo A.

Qué SKU utilizar, en la tabla que aparece a continuación se proporciona un resumen de las características, la capacidad requerida y la SKU específica que se necesita para cada una de ellas.

Escenario	Azure (SKU A)	Office (SKU de P y EM)
Insertar para los clientes (datos en posesión de la aplicación)	✓	✓
Insertar para la organización (datos en posesión del usuario)	✕	✓
Aplicaciones de Microsoft 365 (antes conocidas como aplicaciones de Office 365) • Insertar en Teams • Insertar en SharePoint	✕	✓
Inserción de dirección URL segura (inserción desde el servicio Power BI)	✕	✓

Consideraciones básicas de las capacidades:

	¿Qué es Power BI Embedded de Azure?	Power BI Premium	
Oferta	Azure	Office	
SKU	A	EM	P
Facturación	Cada hora	Mensual	Mensual
Asignación	Ninguno	Anualmente	Mensual o anual
Uso	Los recursos de Azure se pueden: • Escalar vertical y horizontalmente • Pausar y reanudar	Incrustar en aplicaciones y en aplicaciones de Microsoft	Incrustar en aplicaciones, así como en la memoria y capacidad de computación

Tabla resumen de las capacidades de cada tipo de instancia:

Nodos de capacidad	Total de núcleos virtuales	Núcleos virtuales de back-end	RAM (GB)	Núcleos virtuales de front-end	Límites de conexiones dinámicas/DirectQuery (por segundo)	Paralelismo de actualización de modelo
EM1/A1	1	0,5	2.5	0,5	3,75	1
EM2/A2	2	1	5	1	7,5	2
EM3/A3	4	2	10	2	15	3
P1/A4	8	4	25	4	30	6
P2/A5	16	8	50	8	60	12
P3/A6	32	16	100	16	120	24
P4	64	32	200	32	240	48
P5	128	64	400	64	480	96

Para el cálculo de precios de las capacidades Premium tenemos una calculadora. Las diferencias entre EM y P para las SKU Premium son, que las instancias P están enfocadas a la inserción y a las funciones empresariales, que requieren un compromiso mensual o anual, se facturan mensualmente e incluyen una licencia para instalar Power BI Report Server en el entorno local; y las instancias EM están enfocadas a la inserción de la organización, requieren un compromiso anual que se factura mensualmente, las EM1 y EM2 solo están disponibles a través de planes de licencias por volumen y no se pueden comprar directamente.

Para las instancias de Azure se tiene un coste por hora, ajustamos el cálculo de manera aproximada en la siguiente tabla para un mes.

TIPO DE NODO	INFRAESTRUCTURA DEDICADA	NÚCLEOS VIRTUALES	MEMORIA	NÚCLEOS FRONT-END/BACK-END	PRECIO
A1	No	1	3 GB RAM	0,5 / 0,5	~€620,5955/mes
A2	No	2	5 GB RAM	1 / 1	~€1236,2045/mes
A3	Sí	4	10 GB RAM	2 / 2	~€2477,3338/mes
A4	Sí	8	25 GB RAM	4 / 4	~€4959,6540/mes
A5	Sí	16	50 GB RAM	8 / 8	~€9924,2327/mes
A6	Sí	32	100 GB RAM	16 / 16	~€19.853,7597/mes

3. Código

Para este tipo de embedded tenemos un desarrollo de ejemplo hecho con NodeJS, para la autenticación con el tenant de Azure Active Directory usamos adal.

Se incluye a continuación el código para obtener el token de autenticación de AAD:

```
const getAuthenticationToken = async function() {

    // Use ADAL.js for authentication
    let adal = require("adal-node");

    let AuthenticationContext = adal.AuthenticationContext;

    // Create a config variable that store credentials from config.json
    let config = require(__dirname + "/../config/config.json");

    let authorityUrl = config.authorityUri;

    // Check for the MasterUser Authentication
    if (config.authenticationMode.toLowerCase() === "masteruser") {
        let context = new AuthenticationContext(authorityUrl);

        return new Promise(
            (resolve, reject) => {
                context.acquireTokenWithUsernamePassword(config.scope, co
nfig.pbiUsername, config.pbiPassword, config.clientId, function(err, toke
nResponse) {

                    // Function returns error object in tokenResponse
                    // Invalid Username will return empty tokenResponse,
thus err is used
                    if (err) {
                        reject(tokenResponse == null ? err : tokenRespons
e);
                    }
                    resolve(tokenResponse);
                })
            }
        );

        // Check for ServicePrincipal Authentication
    } else if (config.authenticationMode.toLowerCase() === "serviceprinci
pal") {
        authorityUrl = authorityUrl.replace("common", config.tenantId);
        let context = new AuthenticationContext(authorityUrl);
```

```
        return new Promise(
            (resolve, reject) => {
                context.acquireTokenWithClientCredentials(config.scope, c
onfig.clientId, config.clientSecret, function(err, tokenResponse) {

                    // Function returns error object in tokenResponse
                    // Invalid Username will return empty tokenResponse,
thus err is used
                    if (err) {
                        reject(tokenResponse == null ? err : tokenRespons
e);
                    }
                    resolve(tokenResponse);
                })
            }
        );
    }
}
```

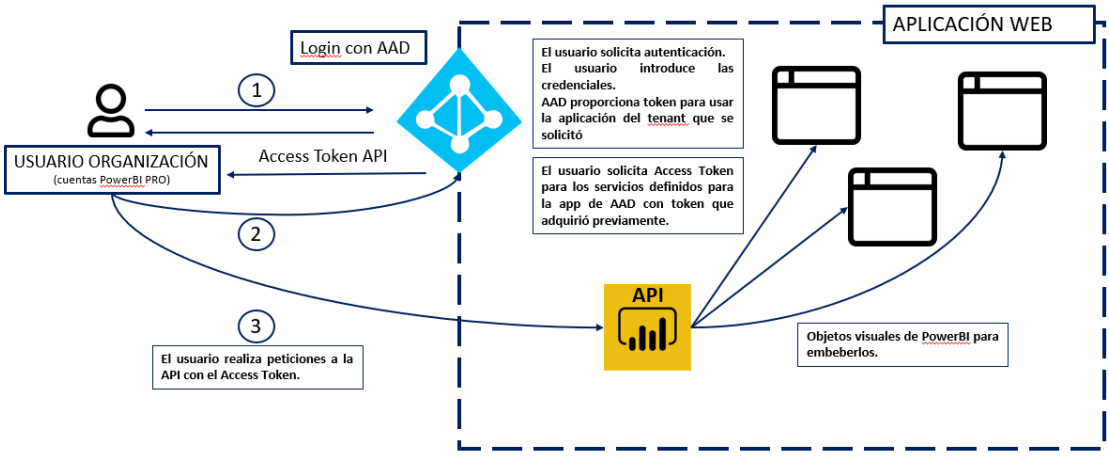
Como podemos ver hemos especificado para este desarrollo "masteruser", esto es una prueba de desarrollo no algo que vayamos a pasar a producción. Este tipo de especificaciones las introducimos en un fichero estático JSON dentro del proyecto, este sería el esquema para este desarrollo:

```
{
    "authenticationMode": "MasterUser",
    "authorityUri": "https://login.microsoftonline.com/common/",
    "scope": "https://analysis.windows.net/powerbi/api",
    "apiUrl": "https://api.powerbi.com/",
    "clientId": "el identificador de nuestra app de AAD",
    "workspaceId": "id workspace donde tenemos el contenido a embeber",
    "reportId": "id report que queremos embeber",
    "pbiUsername": "cuenta del tenant con licencia PowerBI PRO y acceso
al workspace",
    "pbiPassword": "Password de esa cuenta que hará de proxy",
    "clientSecret": "id secreto de la app AAD",
    "tenantId": "id del tenant de AAD"
}
```

Al tener una cuenta PRO tendremos permisos para hacer peticiones a la API y generar tokens de inserción.

4. EMBEDDED PARA ORGANIZACIÓN

1. Esquema de uso PBIE para organización



Para poder usar este tipo de inserción de componentes de PowerBI en nuestra organización necesitamos que cada usuario que acceda tenga cuenta PRO de modo que al logarse genere sus propios tokens.

Con esta opción tenemos disponibles las mismas APIs y SDKs que con el embedded para clientes, tenemos que subir los informes, paneles o visuales a un workspace en ambas opciones también, no obstante, con esta opción no es necesario contratar una capacidad dedicada de Azure o PowerBI Premium.

2. Paso a producción

Como se comentó al final del punto anterior no es necesario instancias de SKU EM, P o A. Se requieren las licencias PRO de PowerBI, un tenant de Azure Active Directory con los usuarios de la organización, una aplicación registrada con los permisos configurados en el tenant y la herramienta PowerBI Desktop para el desarrollo de los informes que se subirán al servicio de PowerBI.

Los precios de las licencias de PowerBI Pro rondan los 8 € al mes por usuario, si ningún usuario externo a la organización necesita acceder a la aplicación que embeberá el contenido de PowerBI, entonces no tendríamos costes adicionales a los mencionados; pero si eventualmente clientes externos van a acceder a estos recursos de PowerBI se tendría que contratar una instancia de PowerBI Premium esto serían capacidades P y EM.

3. Código

Para este tipo de embedded tenemos un desarrollo de ejemplo hecho con React, para la autenticación con el tenant de Azure Active Directory usamos msal.

Se incluye a continuación el código para obtener el token de autenticación de AAD:

```
authenticate(): void {
        const thisObj = this;

        const msalConfig = {
            auth: {
                clientId: config.clientId
            }
        };

        const loginRequest = {
            scopes: config.scopes
        };

        const msalInstance: UserAgentApplication = new UserAgentApplicati
on(msalConfig);

        function successCallback(response: AuthResponse): void {
            let respuestaJSON = JSON.stringify(response)
            if (response.tokenType === "id_token") {
                thisObj.authenticate();

            } else if (response.tokenType === "access_token") {

                accessToken = response.accessToken;
                thisObj.setUsername(response.account.name);
                thisObj.getembedUrl();

            } else {

                thisObj.setState({ error: [("Token type is: " + response.
tokenType)] });
            }
        }

        function failCallBack(error: AuthError): void {

            thisObj.setState({ error: ["Redirect error: " + error] });
        }

        msalInstance.handleRedirectCallback(successCallback, failCallBack
);
```

```
        // check if there is a cached user
        if (msalInstance.getAccount()) {


           // get access token silently from cached id-token
            msalInstance.acquireTokenSilent(loginRequest)
                .then((response: AuthResponse) => {

                    // get access token from response: response.accessTok
en
                    accessToken = response.accessToken;
                    this.setUsername(response.account.name);
                    this.getembedUrl();
                })
                .catch((err: AuthError) => {

                    // refresh access token silently from cached id-token
                    // makes the call to handleredirectcallback
                    if (err.name === "InteractionRequiredAuthError") {
                        msalInstance.acquireTokenRedirect(loginRequest);
                    }
                    else {
                        thisObj.setState({ error: [err.toString()] })
                    }
                });
        } else {

            // user is not logged in or cached, you will need to log them
 in to acquire a token
            msalInstance.loginRedirect(loginRequest);
        }
    }
```

Para este caso no tenemos una cuenta maestra, sino que cada usuario con licencia PRO se logea por si mismo, la última línea de código que vemos es la que nos abrirá el login de Microsoft para introducir nuestras credenciales, es un método propio de la librería msal, adentrándonos en esta librería vemos opciones para guardar el SID y demás métodos para que esta autenticación sea única para todas las aplicaciones del dominio. Por esta razón primero comprobamos que no tengamos cacheado el usuario y evitamos que tenga que reintroducir sus credenciales.

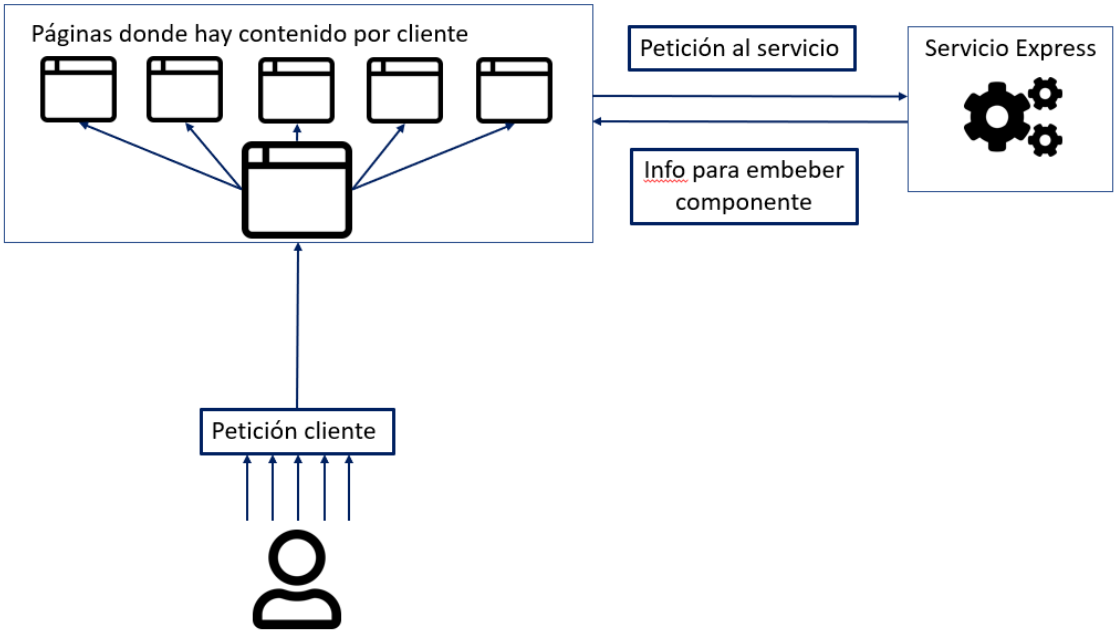
El fichero de configuración es en este caso un fichero TypeScript que exporta las constantes de configuración, se vería de la siguiente manera:

```
export const scopes: string[] = ["https://analysis.windows.net/powerbi/ap
i/Report.Read.All"];

// Client Id (Application Id) of the AAD app.
export const clientId: string = "…";

// Id of the workspace where the report is hosted
export const workspaceId: string = "…";

// Id of the report to be embedded
export const reportId: string = "…";
```

5. FUNCIONAMIENTO ARQUITECTURA

En esta sección se redacta el funcionamiento de nuestra aplicación hecha en NodeJS y su integración con el servicio que desarrolla el cliente .

1. Flujo de información



Este esquema es básico y no se ve nada fuera de la organización, explicaremos a continuación cada parte en detalle incluyendo referencias al código.

2. Petición del cliente

Esto es la petición que los miembros de la organización una vez autenticados hagan mientras navegan por la propia aplicación de cliente.

3. Servicio Express

Desde el servicio desarrollado por el cliente hacemos peticiones ajax a nuestro servicio Express, cada página enviará una petición determinada en función del informe o componente de PowerBI que desea embeber.

La petición será algo como dominio.es/GetEmbedToken:5300?report=proyectosInvestigación devolverá a nuestro cliente la información necesaria para embeber el componente, el cual imaginemos para esta petición es, el report de proyectos de Investigación.

En el ejemplo de la view de general, el js de esa view hace la llamada ajax y utiliza la información para generar una variable de configuración y renderizar el objeto visual de PowerBI.

```
let models = window["powerbi-client"].models;
let reportContainer = $("#report-container").get(0);


// Initialize iframe for embedding report
powerbi.bootstrap(reportContainer, { type: "report" });

$.ajax({
    type: "GET",
    url: "/getEmbedToken",
    dataType: "json",
    data:{"report":"area"},
    success: function(embedData) {

        // Create a config object with type of the object, Embed details
and Token Type
        let reportLoadConfig = {
            type: "report",
            tokenType: models.TokenType.Embed,
            accessToken: embedData.accessToken,
            embedUrl: embedData.embedUrl,
            /*
            // Enable this setting to remove gray shoulders from embedded
 report
            settings: {
                background: models.BackgroundType.Transparent
```

```
            }
            */
        };

        // Use the token expiry to regenerate Embed token for seamless en
d user experience
        // Refer https://aka.ms/RefreshEmbedToken
        tokenExpiry = embedData.expiry;

        // Embed Power BI report when Access token and Embed URL are avai
lable
        let report = powerbi.embed(reportContainer, reportLoadConfig);
```

Nuestro servidor gestiona la petición ajax, cuando recibe ese GET:

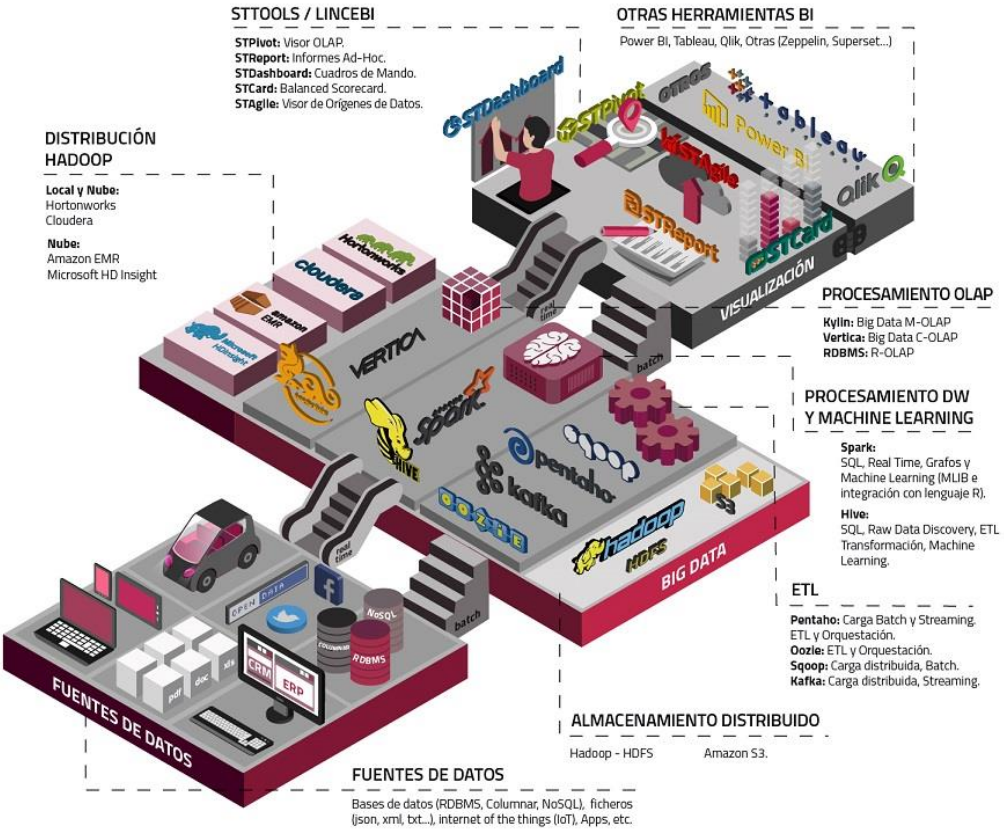
```
app.get('/getEmbedToken', async function(req, res) {
    // Get the details like Embed URL, Access token and Expiry
    let result = await embedToken.generateEmbedToken(req.query.report);

    // result.status specified the statusCode that will be sent along wit
h the result object
    res.status(result.status).send(result);
});
```

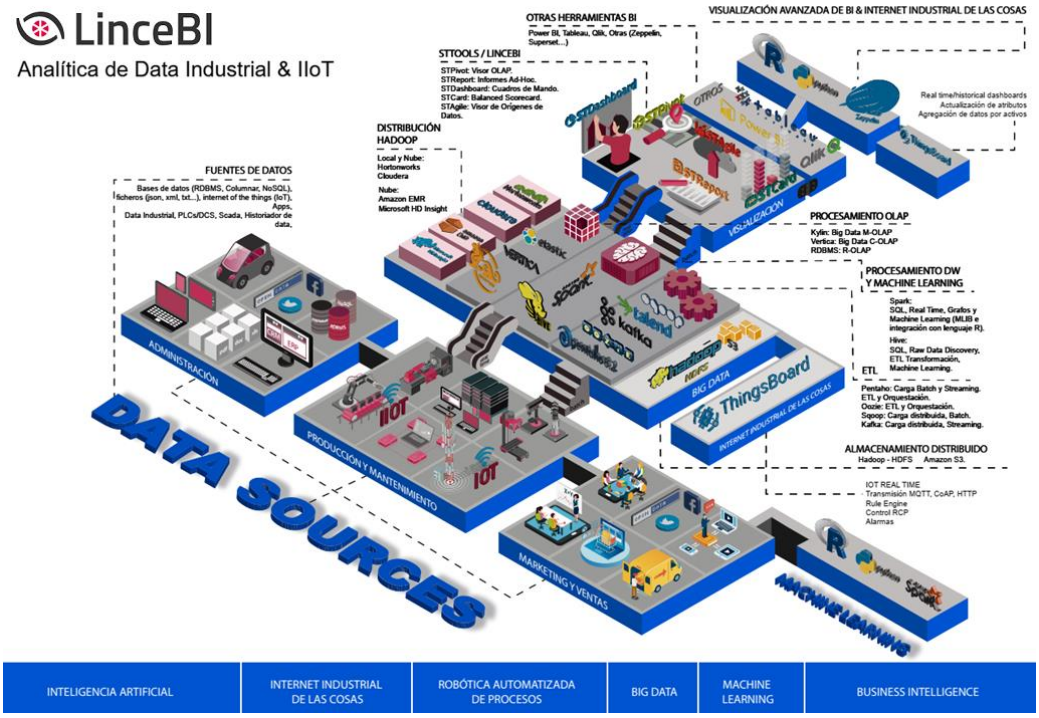
Llama a una función de otro script que se encargará del proceso de autenticación en un primer momento y de las llamadas a la API de PowerBI para obtener la información del componente que queremos embeber.

Necesitamos un archivo JSON estático dentro del servicio que tenga registrados los identificadores de los informes de PowerBI, mediante el query param y un switch seleccionaremos el identificador correspondiente a la página del front (paginas donde hay contenido por cliente) que ha solicitado el cliente.

6. TECNOLOGÍAS

Trabajamos con las principales tecnologías Analytics, siendo Partners Certificados de Vertica, Talend, Microsoft, Snowflake, Kylligence, Pentaho, etc.


STTOOLS / LINCEBI
STPivot: Visor OLAP.
STReport: Informes Ad-Hoc.
STDashboard: Cuadros de Mando.
STCard: Balanced Scorecard.
STAgile: Visor de Orígenes de Datos.
OTRAS HERRAMIENTAS BI
Power BI, Tableau, Qlik, Otras (Zeppelin, Superset...)
DISTRIBUCIÓN HADOOP
Local y Nube:
Hortonworks
Cloudera
Nube:
Amazon EMR
Microsoft HD Insight
VISUALIZACIÓN
PROCESAMIENTO OLAP
Kylin: Big Data M-OLAP
Vertica: Big Data C-OLAP
RDBMS: R-OLAP
PROCESAMIENTO DW Y MACHINE LEARNING
Spark:
SQL, Real Time, Grafos y Machine Learning (MLIB e integración con lenguaje R).
Hive:
SQL, Raw Data Discovery, ETL Transformación, Machine Learning.
BIG DATA
ETL
Pentaho: Carga Batch y Streaming. ETL y Orquestación.
Oozie: ETL y Orquestación.
Sqoop: Carga distribuida, Batch.
Kafka: Carga distribuida, Streaming.
FUENTES DE DATOS
ALMACENAMIENTO DISTRIBUIDO
Hadoop - HDFS
Amazon S3.
FUENTES DE DATOS
Bases de datos (RDBMS, Columnar, NoSQL), ficheros (json, xml, txt...), internet of the things (IoT), Apps, etc.


LinceBI
Analítica de Data Industrial & IIoT
OTRAS HERRAMIENTAS BI
VISUALIZACIÓN AVANZADA DE BI & INTERNET INDUSTRIAL DE LAS COSAS
STTOOLS / LINCEBI
DISTRIBUCIÓN HADOOP
FUENTES DE DATOS
PROCESAMIENTO OLAP
PROCESAMIENTO DW Y MACHINE LEARNING
ETL
ALMACENAMIENTO DISTRIBUIDO
DATA SOURCES
ADMINISTRACIÓN
PRODUCCIÓN Y MANTENIMIENTO
MARKETING Y VENTAS
MACHINE LEARNING
INTELIGENCIA ARTIFICIAL
INTERNET INDUSTRIAL DE LAS COSAS
ROBÓTICA AUTOMATIZADA DE PROCESOS
BIG DATA
MACHINE LEARNING
BUSINESS INTELLIGENCE

7. INFORMACIÓN SOBRE STRATEBI

Stratebi es una empresa española, con sede en Madrid y oficinas en Barcelona, Alicante y Sevilla, con amplia experiencia en sistemas de información, soluciones tecnológicas y procesos relacionados con soluciones de Open Source y de inteligencia de Negocio.

Esta experiencia, adquirida durante la participación en proyectos estratégicos en compañías de reconocido prestigio a nivel internacional, se ha puesto a disposición de nuestros clientes.

Somos **Partners Certificados en Microsoft PowerBI** con una dilatada experiencia

Stratebi es la única empresa española que ha estado presente todos los Pentaho Developers celebrados en Europa habiendo organizado el de España.

En Stratebi nos planteamos como **objetivo** dotar a las compañías e instituciones, de herramientas escalables y adaptadas a sus necesidades, que conformen una estrategia Business Intelligence capaz de rentabilizar la información disponible. Para ello, nos basamos en el desarrollo de soluciones de Inteligencia de Negocio, mediante tecnología Open Source.

Stratebi **son** profesores y responsables de proyectos **del Master en Business Intelligence de la Universidad UOC, UCAM, EOI…**

Los profesionales de Stratebi son los creadores y autores del primer weblog en español sobre el mundo del Business Intelligence, Data Warehouse, CRM, Dashboards, Scorecard y Open Source. Todobi.com

Stratebi es partner de las principales soluciones Analytics: Microsoft Power BI, Talend, Pentaho, Vertica, Snowflake, Kyligence, Cloudera…

Todo Bi, se ha convertido en una referencia para el conocimiento y divulgación del Business Intelligence en español.

8. OTROS

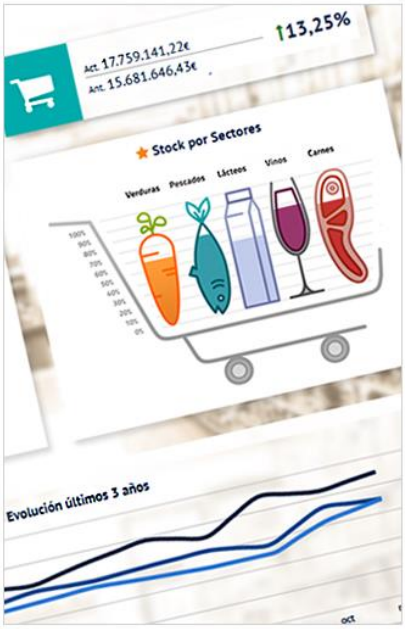
Trabajamos en los principales sectores y con algunas de las compañías y organizaciones más importantes de España.

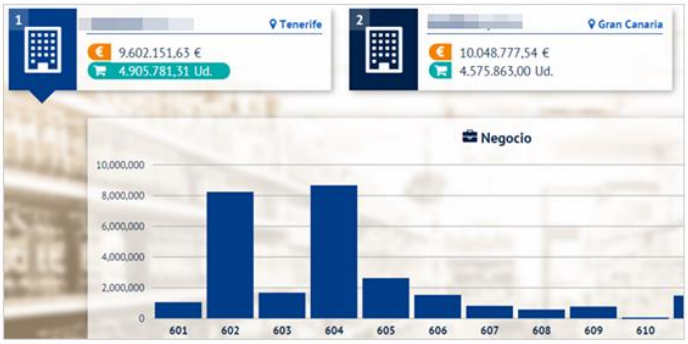
SECTOR PRIVADO

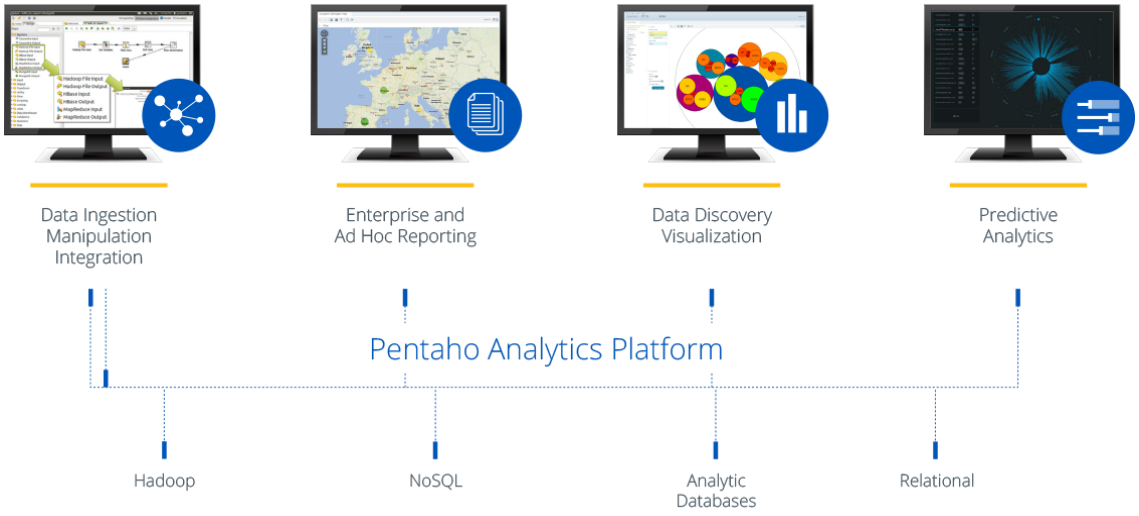
SECTOR PÚBLICO

9. EJEMPLOS DE DESARROLLOS ANALYTICS

A continuación, se presentan **ejemplos de algunos screenshots** de cuadros de mando diseñados por Stratebi, con el fin de dar a conocer lo que se puede llegar a obtener, así como Demos Online en la web de Stratebi:

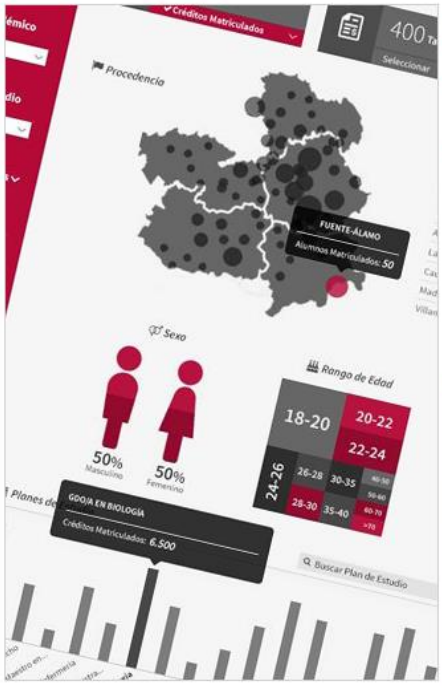




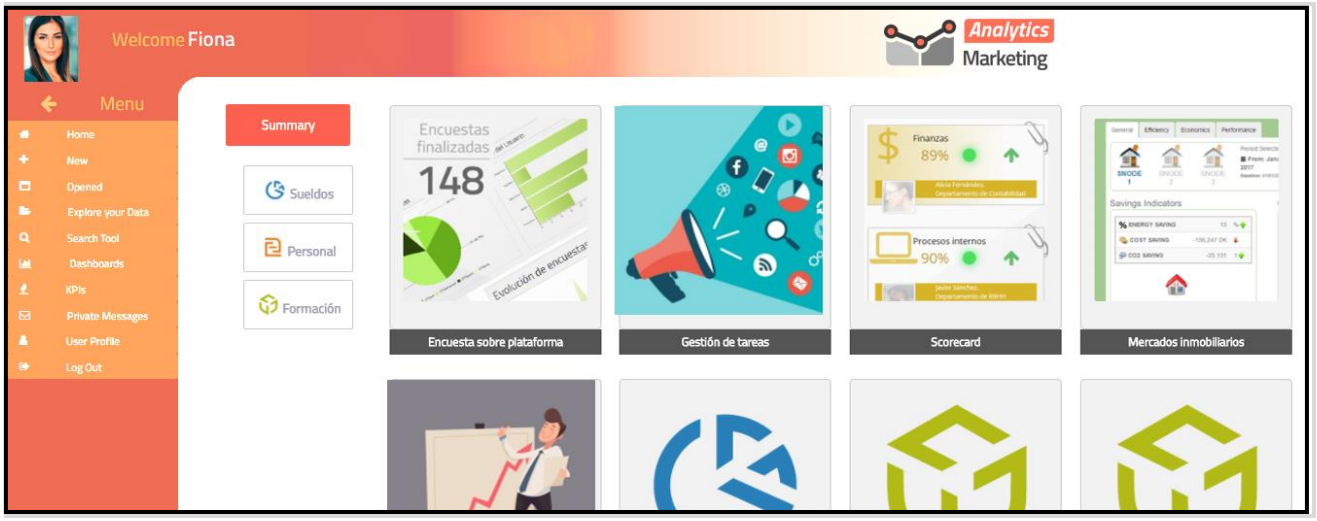


facebook DASHBOARD COMENTARIOS
Sexo
Sentimiento
Localización
65%
35%
28%
60%
12%
Tipo de recurso
Evolución Comentarios por días

smart MADRID
Último Actualización
18:24
TOP 5 PUNTOS DE MEDICIÓN
PROXIMIDAD DE PUNTOS DE MEDIDA
EVOLUCIÓN INTENSIDAD DE TRÁFICO


Créditos Matriculados
Procedencia
FUENTE-ÁLAMO
Alumnos Matriculados: 50
Sexo
50%
50%
Rango de Edad
18-20
20-22
22-24
24-26
26-28
28-30
30-35
35-40
GRADO EN BIOLOGÍA
Créditos Matriculados: 6.500
Buscar Plan de Estudio

181 Inscritos en Cursos
8 Desinscritos en Cursos
181 Inicios de Cursos
110 Inscritos en Eventos Virtuales
63 Inscritos en Debates
181 Inscritos en Talleres
USUARIOS ACTIVOS POR PAÍS
INSCRITOS EN CURSOS POR COMUNIDAD EDU
JORNADA COMPLETA
IDIOMAS
TIC
TIPO DE CENTRO
TIPO DE AUSENCIA
Nº de profesores por sexo
36%
64%
Nº total de profesores
1.587
Media de edad
43 años
Búsqueda


Welcome Fiona
Analytics
Marketing
Menu
Home
New
Opened
Explore your Data
Search Tool
Dashboards
KPIs
Private Messages
User Profile
Log Out
Summary
Sueldos
Personal
Formación
Encuestas finalizadas
148
Evolución de encuestas
Encuesta sobre plataforma
Gestión de tareas
Finanzas
89%
Procesos internos
90%
Scorecard
Savings Indicators
Mercados inmobiliarios