|  |
| --- |
| Workshop de Desarrollo - Laboratorio 01 |
| Apps para Office |
|  |

|  |
| --- |
|  |

# Pre requisitos

Para este laboratorio se requiere que tenga una cuanta activa de Office 365. Si no la tiene puede crear una de prueba por un mes desde la URL <http://office.microsoft.com/es-hn/business/microsoft-office-365-enterprise-e3-software-empresarial-FX103030346.aspx>

## Crear Colección de Sitios para el laboratorio

Una vez aprovisionada la cuenta de Office 365, debe crear una colección de sitios basada en la plantilla “Sitio de desarrollador”. Los pasos son los siguientes:

1. Acceda a su cuenta de Office 365.
2. Acceda a la administración de SharePoint utilizando el menú superior derecho y seleccionando “Administrador 🡪 SharePoint”.
3. En la sección colección de sitios seleccione “Nuevo” en la barra de tareas y luego “Colección de Sitios”.
4. Ingrese los datos de la nueva colección de sitios, seleccione la plantilla “Sitio de Desarrollador” y presione “Aceptar”.

## Configurar el catálogo de Aplicaciones

Si ya tiene una colección de sitios basada en la plantilla “Sitio de desarrollador” proceda a configurar el catálogo de Aplicaciones siguiendo estos pasos en su subscripción Office 365:

1. Acceda a su cuenta de Office 365.
2. Acceda a la administración de SharePoint utilizando el menú superior derecho y seleccionando “Administrador 🡪 SharePoint”
3. Acceda a la sección “Aplicaciones” utilizando la opción de menú que se encuentra en el menú vertical de la izquierda.
4. Acceda a la sección “Catálogo de aplicaciones”. Si usted ya tiene un catálogo configurado, esta opción navega hasta el sitio creado por SharePoint; si no, le brindará la opción para configurar uno.
5. Selecciones la opción “Crear un nuevo sitio de catálogo de aplicaciones” y después “Aceptar”.
6. Ingrese los datos del nuevo Sitio que utilizará para el catálogo de aplicaciones y presione “Aceptar”.
7. El nuevo sitio será aprovisionado en unos minutos y el catálogo de aplicaciones quedará listo para ser utilizado.

# Introducción

## Tiempo estimado para completar este laboratorio

30 a 60 Minutos

## Objetivos

Después de completar este laboratorio usted será capaz de:

* Configurar el catálogo de aplicaciones para una subscripción Office 365 (realizado en los prerrequisitos).
* Realizar aplicaciones utilizando NAPA.
* Realizar aplicaciones utilizando Visual Studio 2013.
* La utilización de JavaScript como lenguaje de programación.
* Realizar Consultas con REST y OData.

## Descripción del Laboratorio

Las aplicaciones de Office utilizan JavaScript, HTML y CSS como lenguajes de programación y tecnologías para la interfaz de usuario respectivamente. En este laboratorio conoceremos los conceptos básicos de uso de estas técnicas de programación, como también dos de las herramientas que podemos utilizar para la construcción de aplicaciones.

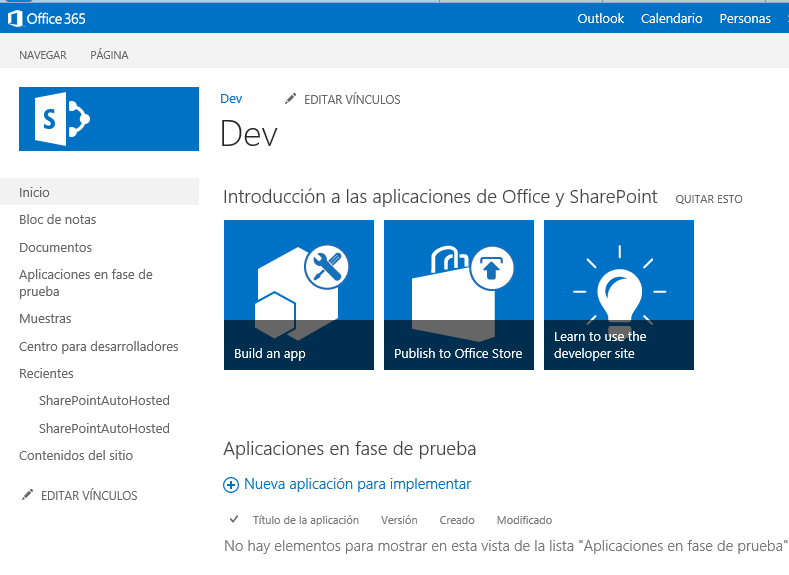
# Ejercicio 1: Mi primer aplicación! con NAPA

En este ejercicio vamos a instalar la aplicación NAPA en nuestra subscripción Office 365 para poder desarrollar aplicaciones.

## Tarea 1 – Instalación de la aplicación NAPA en Office 365

Esta tarea lo guiara en la instalación de la aplicación NAPA en su subscripción de Office 365.

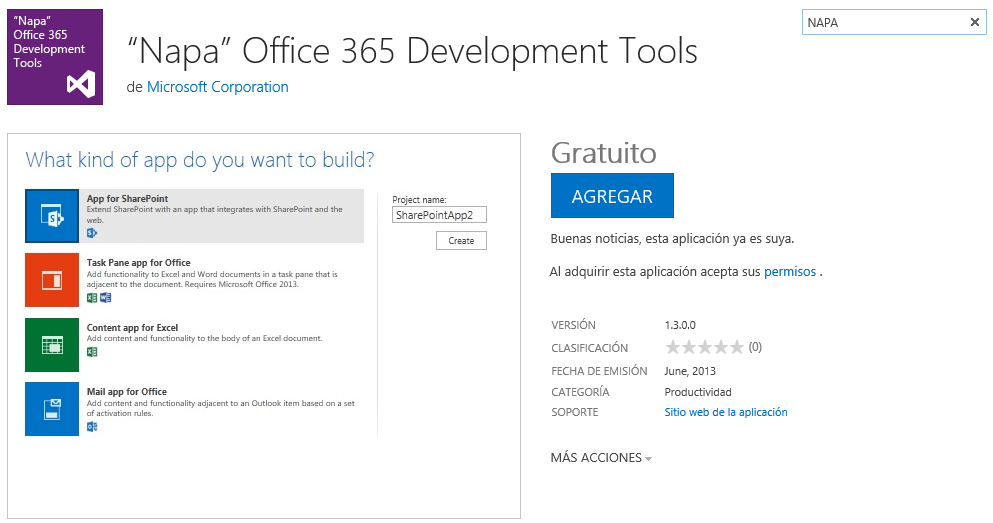
1. Comience la tarea iniciando sesión en su subscripción de Office 365.
2. Acceda al sitio de desarrollador creado anteriormente.
3. En SharePoint 2013 todos los componentes que podemos agregar desde la interfaz de usuario son “Aplicaciones”. Para agregar una nueva acceda a la sección “Contenidos del sitio” que se encuentra en el menú vertical izquierdo como se puede ver en la siguiente imagen.



1. En esa sección encontrará un listado completo de las aplicaciones que se encuentran instaladas en su sitio (Listas, bibliotecas y Sitios). Para instalar una nueva aplicación seleccione “Agregar una Aplicación”, como indica la siguiente imagen:



1. En el menú vertical izquierdo seleccione la opción “Almacén de SharePoint” para acceder a la tiende de aplicaciones de Microsoft. **Nota:** si se produce un error, revise que la configuración del Idioma y Moneda en la parte superior de la página sean los correctos; si no lo están, configúrelos para que la tienda funcione correctamente.
2. Navegue por la tienda para conocer las aplicaciones que se pueden instalar en SharePoint. Busque la aplicación “Napa” en el buscador de la parte superior de la página si no la encontró en la página principal de la tienda.
3. Seleccione la aplicación para navegar a su página principal como se puede ver en la siguiente imagen.



1. Seleccione el botón “Agregar” que posee la aplicación en su página principal y confirme la solicitud de autorización.
2. Una vez finalizado el proceso la aplicación quedará disponible para su utilización.

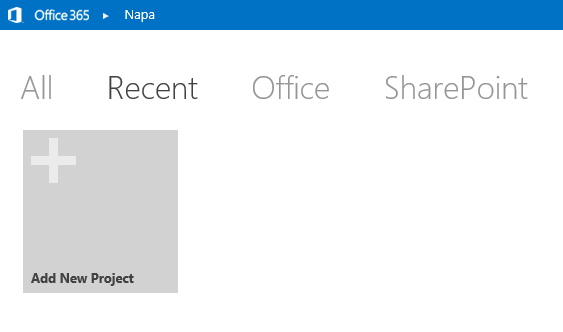
# Ejercicio 2: Crear Aplicación para Office utilizando NAPA.

En este ejercicio se creará una aplicación para Office utilizando NAPA. Una vez que la misma está lista se procederá con su ejecución para ser utilizada en Excel o Word, tanto local como Online.

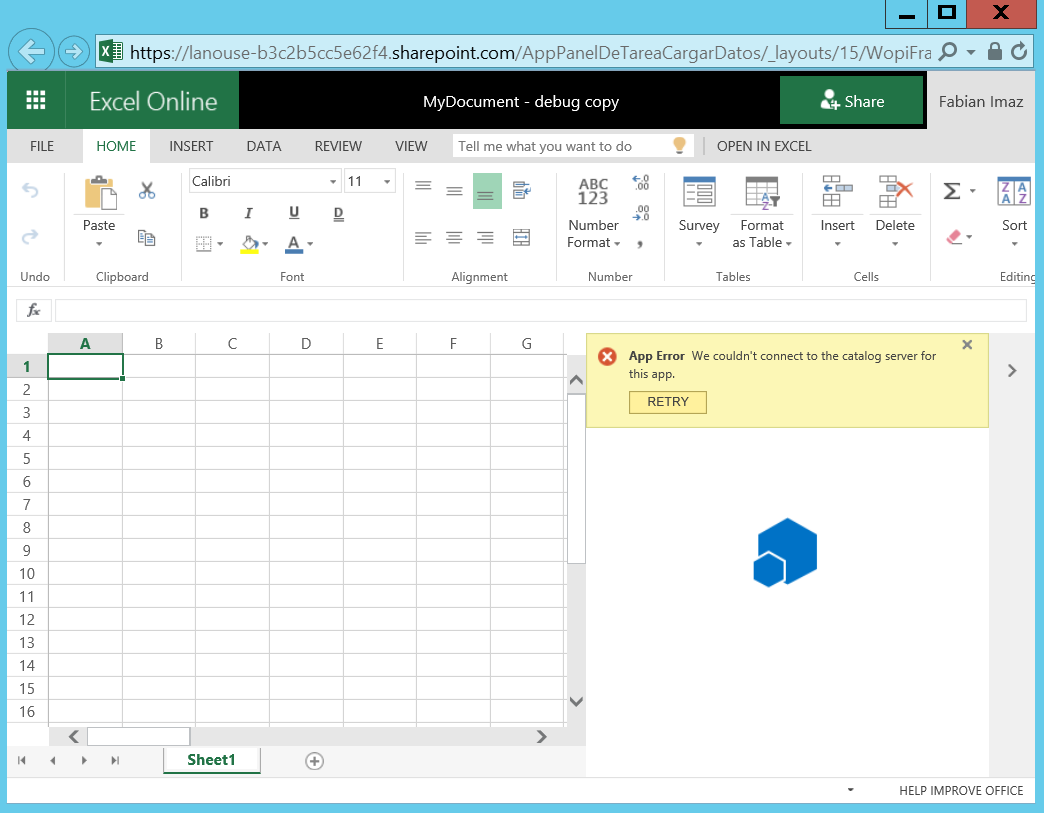
## Tarea 1: Crear una Aplicación para Office con NAPA

En esta tarea creará una aplicación para Office utilizando NAPA.

1. Comience la tarea iniciando sesión en su subscripción de Office 365.
2. Acceda al sitio de desarrollo creado y en la página principal seleccione el Tile “Crear una aplicación” para acceder a la página principal de la aplicación “Napa” como se puede ver en la siguiente Imagen.



1. Seleccione el Tile “Add New Project” para crear una nueva aplicación.
2. En la ventana emergente seleccione “Task Pane for Office”, introduzca el siguiente texto “AppPanelDeTareaCargarDatos” en el cuadro “Project Name” y presione el botón “Create” para crear la aplicación.
3. Una vez finalizado el proceso de creación, “NAPA” abre el proyecto y navega hasta su página principal. En el menú izquierdo seleccione “Home.js” para revisar el código JavaScript que tiene implementada la aplicación por defecto, como podrá apreciar en el archivo “App.js” se encuentra el código para el manejo del panel informativo y en el archivo “Home.js” se encuentra el código para interactuar con el documento Office.
4. Seleccione el ícono  “Run Project” que se encuentra en el menú inferior izquierdo para probar la aplicación. Se abrirá una ventana modal que mostrará el proceso de armado del paquete, instalación y ejecución, al terminará se abrirá una nueva ventana cargando un documento Excel en el navegador. También puede abrir el documento seleccionando el enlace “Click here to launch your app in a new window” que se encuentra en la ventana modal desplegada anteriormente que le permite cargar el documento Excel con la aplicación de panel embebida en el mismo.



1. Seleccione una celda en el documento Excel, introduzca un texto y luego en la aplicación seleccione el botón “**XXXXXXXX**”. El texto será copiado de la celda al panel de notificaciones que se creó en la aplicación.

## Tarea 2: Agregar una consulta REST que cargue una tabla con datos.

Siga los pasos que se indican a continuación para agregar una consulta REST y cargar los datos que se devuelven en una tabla.

En esta tarea agregará los controles HTML en la página principal de la aplicación y el código correspondiente para extraer datos de una base de datos expuesta utilizando REST y cargar los resultados en una tabla dentro del documento Excel.

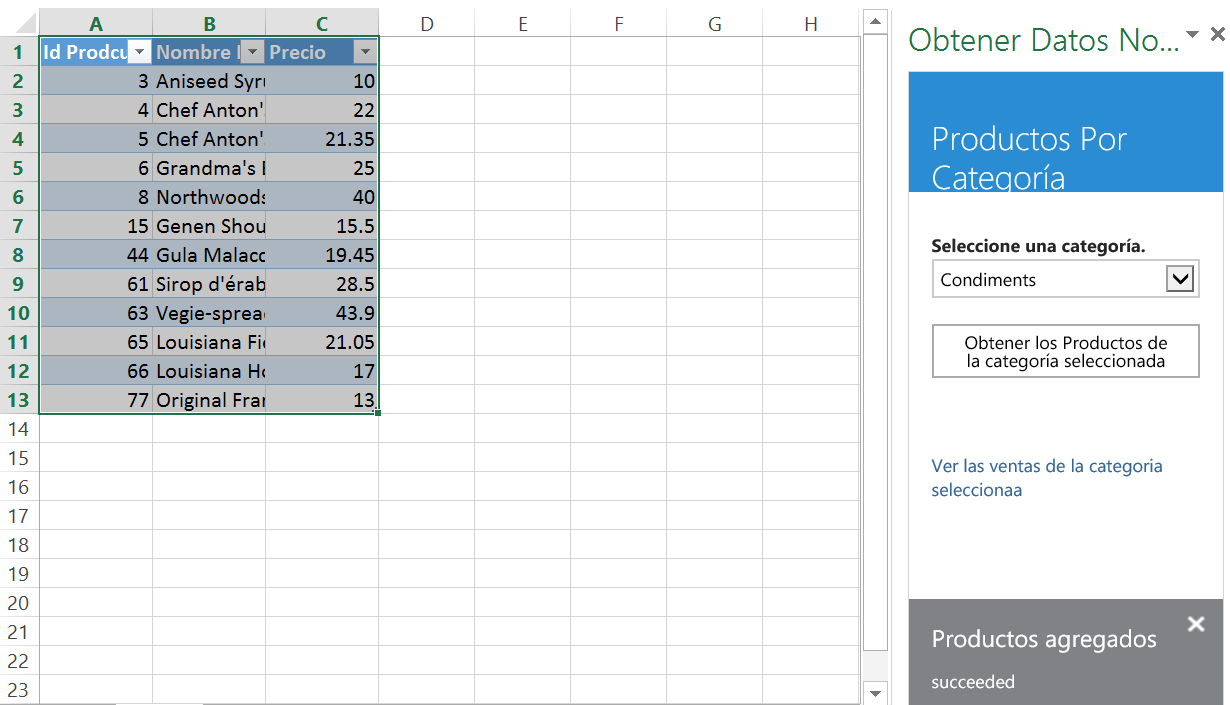
1. Comience la tarea iniciando sesión en su subscripción de Office 365.
2. Acceda al sitio de desarrollo creado y en la página principal seleccione el Tile “Build an app” para acceder a la página principal de la aplicación “Napa”.
3. En la página principal seleccione la aplicación creada anteriormente “AppPanelDeTareaCargarDatos”.
4. Acceda al archivo “Home.html” que se encuentra en la sección “Home” en el menú izquierdo.
5. Seleccione todo el contenido del archivo y bórrelo.
6. Copie y pegue todo el código de la siguiente sección en el archivo “Home.html”.

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>  <meta charset="UTF-8" />  <meta http-equiv="X-UA-Compatible" content="IE=Edge" />  <title>AppPanelDeTareaCargarDatos</title>  <script src="../../Scripts/jquery-1.8.2.js" type="text/javascript"></script>  <link href="../../Content/Office.css" rel="stylesheet" type="text/css" />  <script src="https://appsforoffice.microsoft.com/lib/1.0/hosted/office.js" type="text/javascript"></script>  <!-- Para habilitar la depuración sin conexión mediante una referencia local a Office.js, use: -->  <!-- <script src="../../Scripts/Office/MicrosoftAjax.js" type="text/javascript"></script> -->  <!-- <script src="../../Scripts/Office/1.0/office.js" type="text/javascript"></script> -->  <link href="../App.css" rel="stylesheet" type="text/css" />  <script src="../App.js" type="text/javascript"></script>  <link href="Home.css" rel="stylesheet" type="text/css" />  <script src="Home.js" type="text/javascript"></script>  </head>  <body>  <!-- Contenido de página -->  <div id="content-header">  <div class="padding">  <h1>Productos Por Categoría</h1>  </div>  </div>  <div id="content-main">  <div class="padding">  <p><strong>Seleccione una categoría.</strong>  <select id="slCategorias">  <option>Seleccione una categoría</option>  </select></p>  <button id="get-data-productos-categoria">Obtener los Productos de la categoría seleccionada</button>  <p style="margin-top: 50px;">  <a href="../VentasPorCategoria/VantasPorCategoria.html">Ver las ventas de la categoria seleccionaa</a>  </p>  </div>  </div>  </body>  </html> |

1. Revise el código el cual agrega un control del tipo “Select” donde se cargará dinámicamente la lista de categorías de productos disponibles en la base de datos, un botón que al seleccionarlo obtendrá la categoría seleccionada y cargara la lista de productos en el documento Office.
2. En el menú izquierdo seleccione el archivo “Home.js” que se encuentra en la sección “Home” en el menú izquierdo.
3. Seleccione todo el contenido del archivo y bórrelo.
4. Copie y pegue todo el código de la siguiente sección en el archivo “Home.js”.

|  |
| --- |
| /// <reference path="../App.js" />  /\*global app\*/  (function () {  "use strict";  var mIdCategoria = '';  // La función de inicialización se debe ejecutar cada vez que se cargue una página nueva  Office.initialize = function (reason) {  $(document).ready(function () {  app.initialize();  $('#get-data-productos-categoria').click(getDataProductosCategoriesNorthwind);  $('#slCategorias').change(getIdCategorySelected);  getCategoriasNorthwind();  });  };  //Carga las categorias desde el servicio REST  function getCategoriasNorthwind() {  $.ajax({  url: "http://services.odata.org/Northwind/Northwind.svc/Categories?$select=CategoryID,CategoryName&$format=json&$callback=?",  contentType: 'application/json; charset=utf-8',  dataType: 'jsonp',  type: "GET",  success: getCategoriesNorthwindSucess,  error: function (pRequest, pExito, pError) {  PrintMessageNotification(pError, pExito);  }  });  }  //Procesa los datos que son recibidos de la consulta REST  function getCategoriesNorthwindSucess(pDatos, pExito) {  PrintMessageNotification("Cartegorias", pExito);  $(pDatos.value).each(function (clave, valor) {  $('#slCategorias')  .append($('<option>', { value: valor.CategoryID })  .text(valor.CategoryName));  });  }  //obtiene los productos de la categoria seleccionada y los carga en el Excel  function getDataProductosCategoriesNorthwind() {  $('#notification-message').hide();  getIdCategorySelected();  var lUrl = "http://services.odata.org/Northwind/Northwind.svc/Products?$select=ProductID,ProductName,UnitPrice&$filter=CategoryID eq " + mIdCategoria + "&$format=json&$callback=?"  $.ajax({  url: lUrl,  contentType: 'application/json; charset=utf-8',  dataType: 'jsonp',  type: "GET",  success: SetProductosDocumentoExcelSuccess,  error: function (pRequest, pExito, pError) {  PrintMessageNotification(pError, pExito);  }  });  }  function SetProductosDocumentoExcelSuccess(pDatos, pExito) {  var lTabla = new Office.TableData();  lTabla.headers = ["Id Prodcuto", "Nombre Producto", "Precio"];  var lRows = new Array();  lRows = ArmarFilasTabla(pDatos.value);  lTabla.rows = lRows;  // se persite la categoría que el usuario selecciono  GuardarIdCategoria(mIdCategoria);  Office.context.document.bindings.getByIdAsync("Productos", function (pResultado) {  if (pResultado.status == Office.AsyncResultStatus.Succeeded) {  //borrar las filas e insertar las filas  pResultado.value.deleteAllDataValuesAsync();  //se cargan los datos nuevos traidos por el servicio  pResultado.value.addRowsAsync(lRows, function (pResultado) {  if (pResultado.status == Office.AsyncResultStatus.Succeeded) {  PrintMessageNotification("Productos sincronizados", pResultado.status);  }  else {  PrintMessageNotification("Productos sincronizados", pResultado.status);  }  });  }  else {  Office.context.document.setSelectedDataAsync(lTabla, function (pResultado) {  if (pResultado.status == Office.AsyncResultStatus.Succeeded) {  Office.context.document.bindings.addFromSelectionAsync(Office.BindingType.Table, { id: "Productos" }, function (pResultado) {  if (pResultado.status == Office.AsyncResultStatus.Succeeded) {  PrintMessageNotification("Productos agregados", pResultado.status);  }  else {  PrintMessageNotification("Productos no agregados", pResultado.status);  }  });  }  else {  //no se puedo agregar la tabla.  PrintMessageNotification("Error: " + pResultado.error.name, pResultado.error.message);  }  });  }  });  }  //se arma la tabla a partir de los datos recibidos del servicio  function ArmarFilasTabla(pDatos) {  var lRows = new Array();  $.each(pDatos, function (pIndice, pValor) {  lRows[pIndice] = [pValor.ProductID, pValor.ProductName, pValor.UnitPrice];  });  return lRows;  }  //Guardar la categoría seleccionada por el usuario  function GuardarIdCategoria(pIdCategoria) {  Office.context.document.settings.set("IdDeLaCategoria", mIdCategoria);  Office.context.document.settings.saveAsync();  }  //Obtiene el Id de la categoría seleccionada.  function getIdCategorySelected() {  var lIdCatAux = $("#slCategorias").val();  if ($.isNumeric(lIdCatAux)) {  mIdCategoria = lIdCatAux;  }  else {  mIdCategoria = null;  }  }  //Muestra un mensanje de error  function PrintMessageNotification(pTitulo, pMsg) {  app.showNotification(pTitulo, pMsg);  }  })(); |

1. Seleccione el ícono  “Run Project” que se encuentra en el menú inferior izquierdo para probar la aplicación. Se abrirá una ventana modal que mostrará el proceso de armado del paquete, instalación y ejecución, al terminará se abrirá una nueva ventana cargando un documento Excel en el navegador.
2. En la aplicación seleccione un categoría y luego presione el botón “Obtener los Productos de la categoría seleccionada” para cargar la lista de productos en el documento Office como se puede ver en la siguiente imagen.



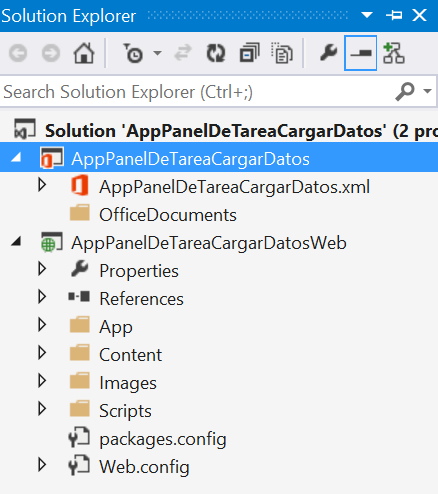
## Tarea 3: Exportar la aplicación para abrirla en Visual Studio.

En esta tarea se exportará la aplicación de Office realizada en NAPA para abrirla en Visual Studio y continuar con el desarrollo.

1. Comience la tarea iniciando sesión en su subscripción de Office 365.
2. Acceda al sitio de desarrollo creado y en la página principal seleccione el Tile “Build an app” para accqeder a la página principal de la aplicación “Napa”.
3. Seleccione la aplicación “AppPanelDeTareaCargarDatos” para abrirla.
4. Seleccione el ícono  “Open in Visual Studio” en el menú izquierdo.
5. Guarde el archivo con extensión “.exe” en un directorio de su computadora, por ejemplo “C:\AppPanelDeTareaCargarDatos”.
6. Inicie el “Explorador de Archivos”, navegue hasta la carpeta donde guardo el archivo “.exe” y ejecútelo como administrador.

**Nota:** Puede producirse un error debido a que falte un complemento en Visual Studio, si este fuese el caso, instálelo cuando se le solicita y al terminar vuelva a ejecutar el archivo.

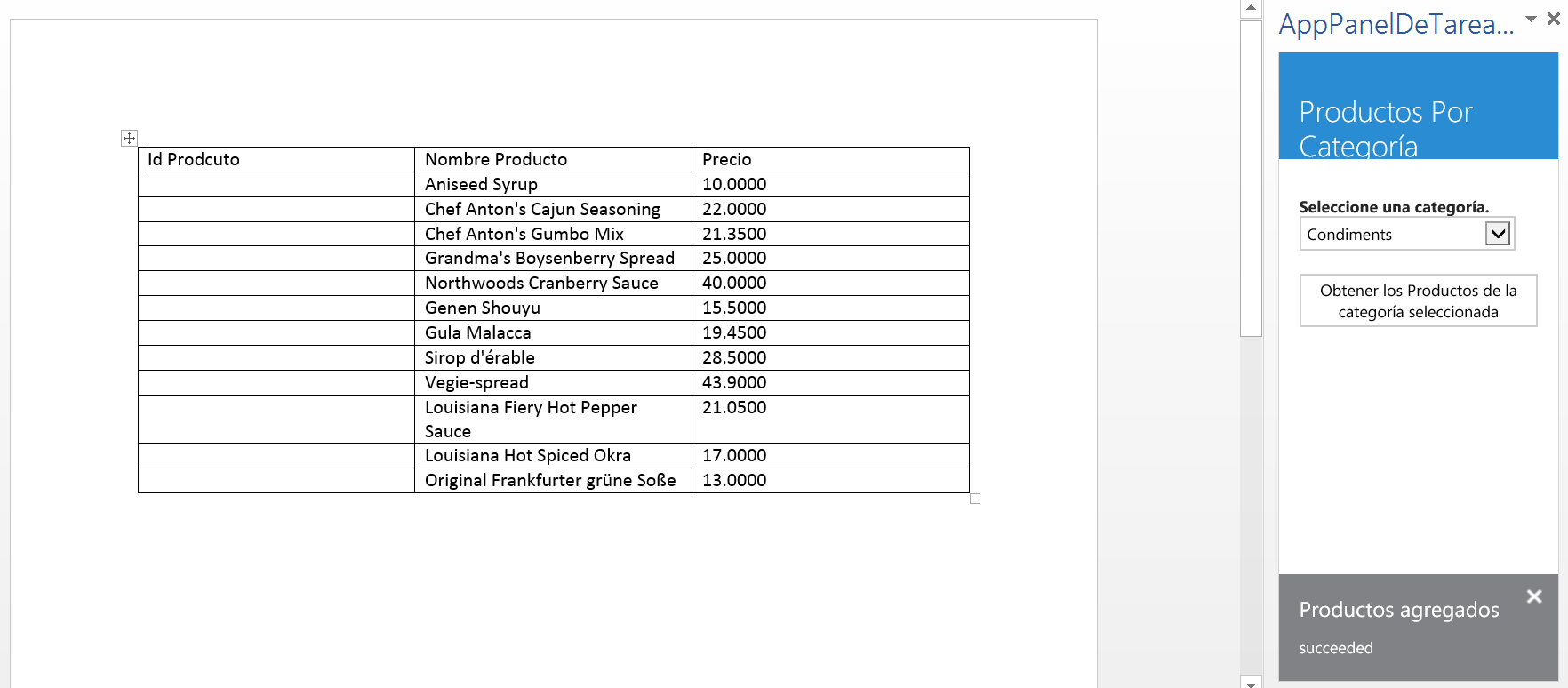
1. Se iniciara Visual Studio y la aplicación será cargada. Se crearán dos proyectos para la misma aplicación, un proyecto de “Aplicación Office” que solamente contendrá el manifiesto y un proyecto Web que contendrá todos los archivos de la aplicación.
2. Acceda al “Explorador de Soluciones” de Visual Studio y analice como quedo la aplicación importada, se tuvo que haber creado una estructura similar a la que se muestra en la siguiente imagen.



1. Acceda al “Explorador de Soluciones”, expanda el proyecto web “AppPanelDeTareaCargarDatosWeb”, expanda la carpeta “App”, luego la carpeta “Home” y por último abra el archivo “Home.html”.
2. Cambie la referencia a la librería jQuery que se encuentra en la línea 8 por la que se encuentra en la siguiente sección.

|  |
| --- |
| <script src="http://code.jquery.com/jquery-1.9.1.js" type="text/javascript"></script> |

1. En la barra superior de herramientas de Visual Studio seleccione él icono  “Start” para ejecutar la aplicación. Se abrirá por defecto un documento Excel en el cliente Office instalado y se cargará la aplicación.
2. Después de probar la aplicación cierre Excel y vuelva nuevamente a Visual Studio.
3. Acceda al “Explorador de Soluciones”, expanda el proyecto de aplicación Office “AppPanelDeTareaCargarDatos”, seleccione el manifiesto con extensión “XML”, presione el botón secundario del ratón y luego seleccione la opción “Abrir”.
4. Acceda a la sección “Activación”, en la sección “Aplicaciones” desmarque todas las opciones menos la correspondiente a “Word”, guarde los cambios y vuelva a ejecutar la aplicación nuevamente.
5. Ejecute nuevamente la aplicación en Visual Studio y en está ocasión se abrirá un documento “Word”. Pruebe nuevamente la aplicación seleccionando una categoría y presione el botón para insertar los productos en el documento como se puede ver en la siguiente imagen.



## Tarea 4: Agregar una nueva página en la aplicación.

En esta tarea se incorporara una nueva página HTML a la aplicación y se vinculara desde la página principal.

1. Acceda a Visual Studio, seleccione el proyecto Web “AppPanelDeTareaCargarDatosWeb” y expándalo para acceder a su contenido.
2. Seleccione la carpeta “App”, presione el botón secundario del ratón, seleccione la opción “Agregar -> Nueva Carpeta” y digite “VentasPorCategoria” como nombre de la misma.
3. Seleccione la carpeta recién creada, presione el botón secundario del ratón, y seleccione la opción “Agregar -> Nuevo Elemento”.
4. En la nueva ventana abierta por Visual Studio seleccione la plantilla “Pagina HTML”, en el cuadro correspondiente al nombre ingrese “VantasPorCategoria” y luego presione el botón “Agregar”.
5. Seleccione todo el contenido del archivo recién creado y bórrelo.
6. Copie y pegue todo el código de la siguiente sección en el archivo “VantasPorCategoria.html”.

|  |
| --- |
| <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml">  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <meta charset="UTF-8" />  <meta http-equiv="X-UA-Compatible" content="IE=Edge" />  <title></title>  <!--<script src="../../Scripts/jquery-1.8.2.js" type="text/javascript"></script>-->  <script src="http://code.jquery.com/jquery-1.9.1.js" type="text/javascript"></script>  <link href="../../Content/Office.css" rel="stylesheet" type="text/css" />  <script src="https://appsforoffice.microsoft.com/lib/1.0/hosted/office.js" type="text/javascript"></script>  <!-- Para habilitar la depuración sin conexión mediante una referencia local a Office.js, use: -->  <!-- <script src="../../Scripts/Office/MicrosoftAjax.js" type="text/javascript"></script> -->  <!-- <script src="../../Scripts/Office/1.0/office.js" type="text/javascript"></script> -->  <link href="../App.css" rel="stylesheet" type="text/css" />  <script src="../App.js" type="text/javascript"></script>  <script src="VentasPorCategoria.js" type="text/javascript"></script>  </head>  <body>  <!-- Contenido de página -->  <div id="content-header">  <div class="padding">  <h1>Ventas Por Categoría</h1>  </div>  </div>  <div id="content-main">  <div class="padding">  <table style="width: 100%;" id="vtaCategoria">  <thead>  <tr>  <td>Nombre Producto</td>  <td>Ventas</td>  </tr>  </thead>  <tbody></tbody>  </table>  <p></p>  <p style="margin-top: 50px;">  <a href="../Home/Home.html">Volver al Inicio</a>  </p>  </div>  </div>  </body>  </html> |

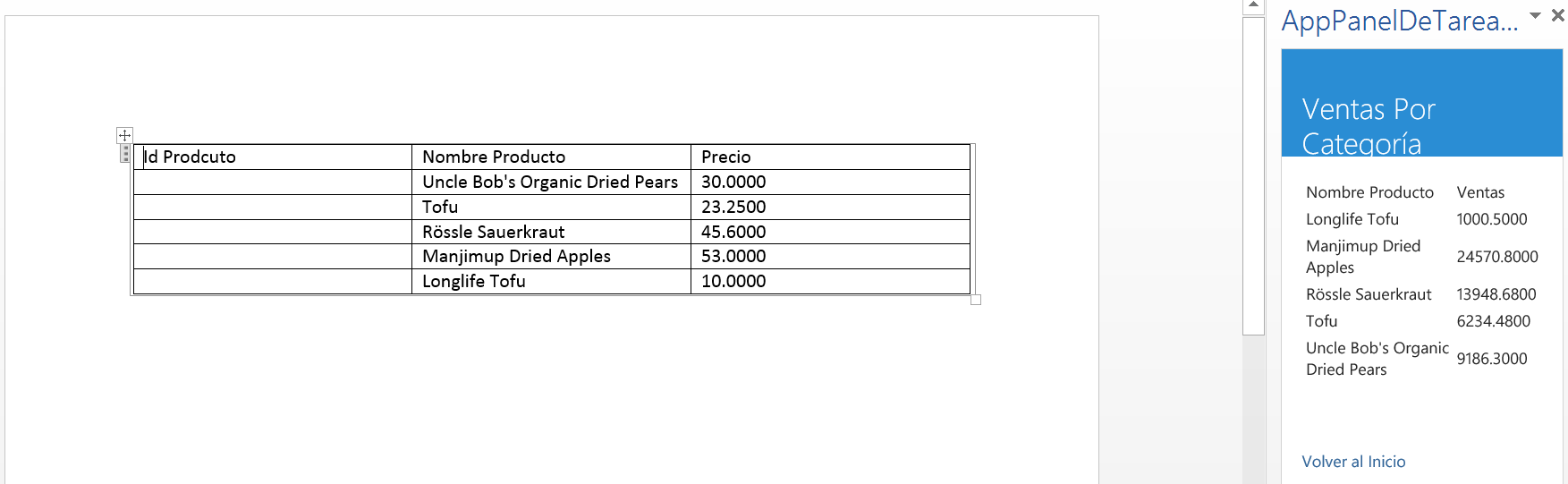
1. Revise el código el cual agrega una tabla HTML que se cargará dinámicamente con las ventas realizadas.
2. En el “Explorador de Soluciones” seleccione la carpeta “VentasPorCategoria” nuevamente, presione el botón secundario del ratón y seleccione la opción “Agregar -> Nuevo Elemento”.
3. En la nueva ventana abierta por Visual Studio seleccione la plantilla “Archivo JavaScript”, en el cuadro correspondiente al nombre ingrese “VentasPorCategoria” y luego presione el botón “Agregar”.
4. Seleccione todo el contenido del archivo recién creado y bórrelo.
5. Copie y pegue todo el código de la siguiente sección en el archivo “VentasPorCategoria.js”.

|  |
| --- |
| /// <reference path="../App.js" />  // La función de inicialización se debe ejecutar cada vez que se cargue una página nueva  Office.initialize = function (reason) {  $(document).ready(function () {  app.initialize();  ObtenerVentasCategoria();  });  };  //Obtiene las ventas de la catetoria desde la base de datos  function ObtenerVentasCategoria() {  var lIdeCategoria = ObtenerIdCategoria();  var lUrl = "http://services.odata.org/Northwind/Northwind.svc/Sales\_by\_Categories?$select=ProductName,ProductSales&$filter=CategoryID eq " + lIdeCategoria + "&$format=json&$callback=?"  $.ajax({  url: lUrl,  contentType: 'application/json; charset=utf-8',  dataType: 'jsonp',  type: "GET",  success: CargarVentasCategoriaSuccess,  error: function (pRequest, pExito, pError) {  PrintMessageNotification(pError, pExito);  }  });  }  //Carga la tabla de precios de la cartegoria  function CargarVentasCategoriaSuccess(pDatos, pExito) {  $(pDatos.value).each(function (clave, valor) {  $('#vtaCategoria').find('tbody').append($('<tr><td>' + valor.ProductName + '</td><td>' + valor.ProductSales + '</td></tr>'));  });  }  //Obtiene el id de la categoria almacenada en el documento  function ObtenerIdCategoria() {  return Office.context.document.settings.get("IdDeLaCategoria");  }  //Muestra un mensanje de error  function PrintMessageNotification(pTitulo, pMsg) {  app.showNotification(pTitulo, pMsg)  } |

1. Revise el código JavaScript recientemente incorporado, el cual obtiene el Id de la categoría seleccionada en la página principal y armar la consulta REST utilizando OData para cargar las ventas realizadas para dicha categoría.
2. El último paso en configurar la navegación desde la página principal hacía la nueva página recién creada, para eso se agregar un enlace en la Home. Acceda al “Explorador de Soluciones”, en el proyecto Web seleccione la capeta “App”, expanda la carpeta “Home” y abra el archivo “Home.html”.
3. Copie y pegue el código de la siguiente sección inmediatamente del elemento HTML <button> que representa el botón en la página principal de la aplicación

|  |
| --- |
| <p style="margin-top: 50px;">  <a href="../VentasPorCategoria/VantasPorCategoria.html">Ver las ventas de la categoria seleccionaa</a>  </p> |

1. En la barra superior de herramientas de Visual Studio seleccione él icono  “Start” para ejecutar la aplicación, cuando esta termine de cargarse seleccione una categoría, presione el botón “Obtener los Productos de la categoría seleccionada” que carga la lista de productos en el documento guardando el Id de la categoría y luego seleccione el enlace “Ver las ventas de la categoría seleccionada” para desplegar la nueva página que desplegará todas las ventas de la misma como se puede ver en la siguiente imagen.



# Resumen del laboratorio

En el anterior laboratorio fue posible ejercitarse en:

* Configurar el sitio de desarrollo y el catálogo de aplicaciones.
* Instalar y usar la aplicación NAPA.
* Crear aplicaciones para Office.
* Programar utilizando jQuery, JavaScript, REST y OData.