



Área de Ingeniería en Computadores

Bases de Datos

Prof. Marco Rivera

Documentación.

Proyecto II Bases de Datos

Estudiantes:

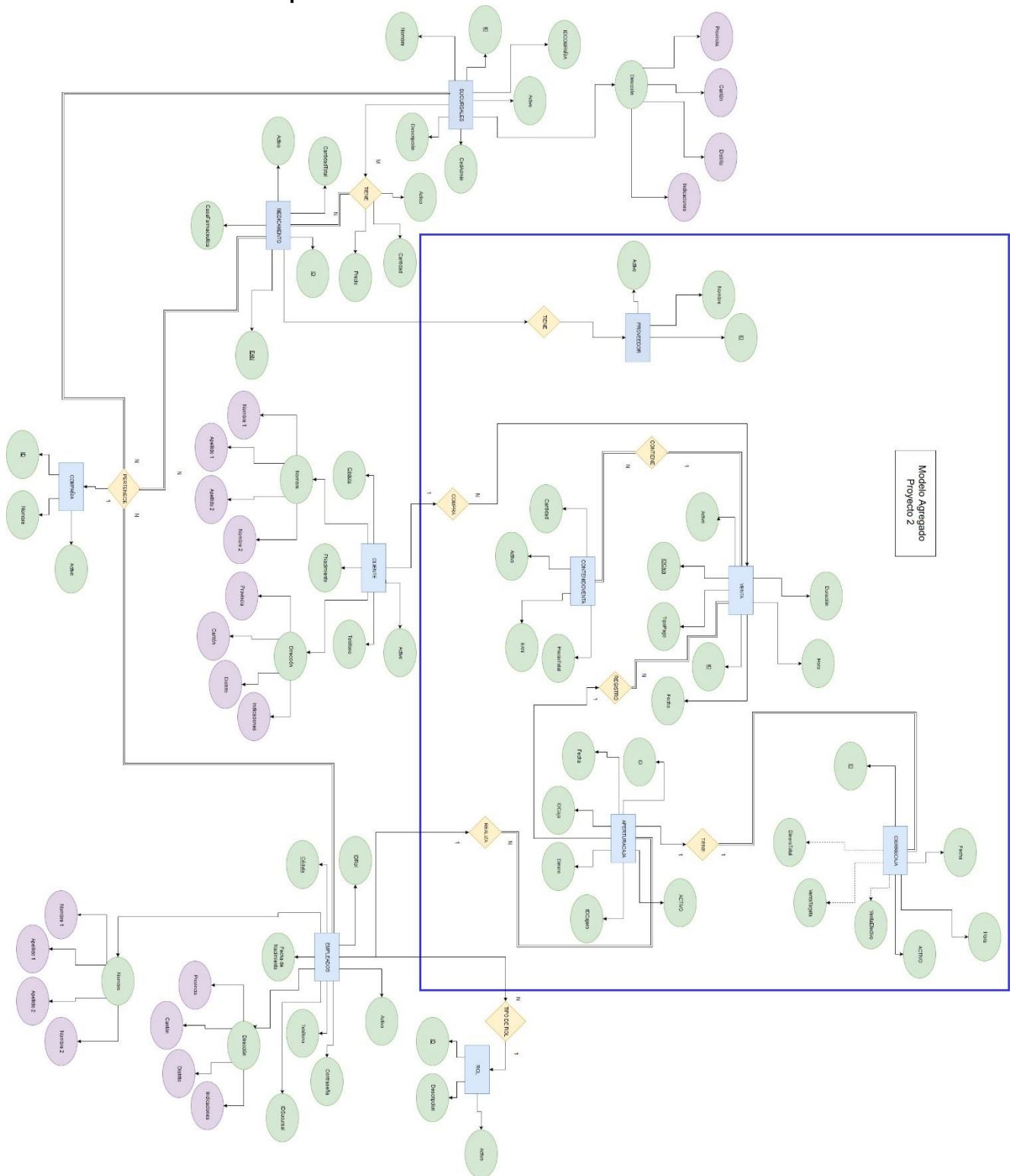
| | |
|-------------------------|------------|
| Efrén Carvajal Valverde | 2015019679 |
| Diego Solís Jiménez | 2014027244 |
| Kevin Acuña Mena | 2015093739 |
| Luis Fernando Murillo | 2015088157 |

Semestre 2 – 2017

Contenido

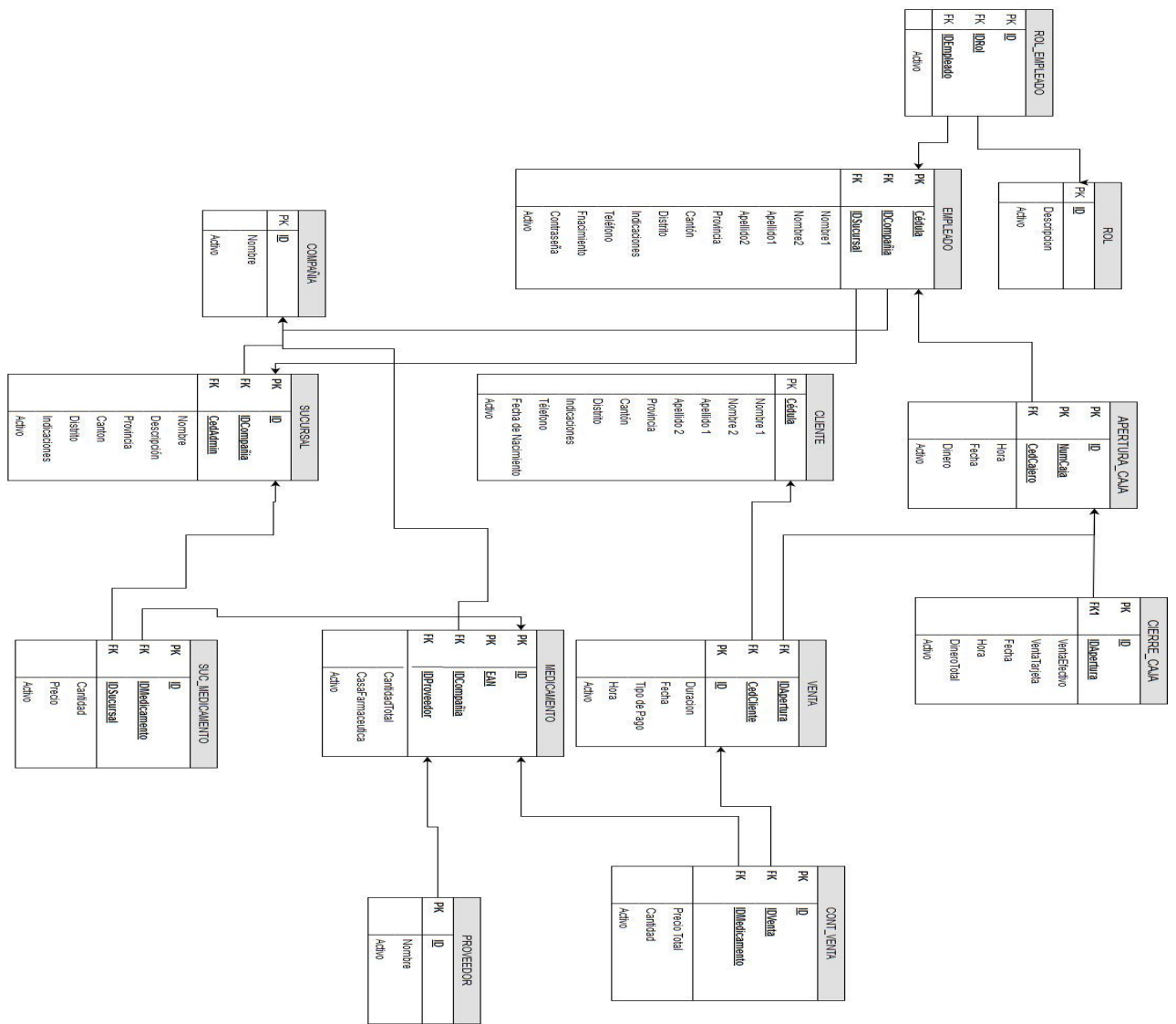
| | |
|---|----|
| 1. Modelo Conceptual. | 3 |
| 2. Modelo Relacional. | 4 |
| 3. Descripción de los métodos y algoritmos implementados. | 5 |
| 4. Descripción detallada de las Estructuras de datos Utilizadas. | 9 |
| 5. Problemas Conocidos. | 10 |
| 6. Problemas Encontrados. | 11 |
| 7. Recomendaciones y Conclusiones. | 11 |
| • Recomendaciones. | 12 |
| • Conclusiones. | 12 |
| 8. Bibliografía. | 13 |

1. Modelo Conceptual.



(Figura 1. Modelo Relacional.)

2. Modelo Relacional.



(Figura 2. Modelo Relacional.)

3. Descripción de los métodos y algoritmos implementados.

public class Caja:

Public **JSONObject** CierreCaja(**JSONObject** x):

Este método toma la entrada “x”, la cual corresponde al ID de la apertura de la caja (IDApertura), y realiza un cierre de caja siempre y cuando no haya sido cerrada la caja después de la última apertura (O sea, no puede haber dos cierres para una apertura).

Para esto, realiza una suma de los precios totales en cada contenido de venta, para cada venta asociada a la última apertura de esta caja, dependiendo de si la venta fue hecha en “Efectivo” o en “Tarjeta” a las variables correspondientes, luego calcula el dinero en efectivo total que queda en la caja sumando el total de las ventas en efectivo con el dinero inicial de la caja. Posteriormente genera las inserciones a ambas bases y guarda los cambios.

public **JSONObject** AperturaCaja(**JSONObject** x):

Realiza una inserción la tabla apertura_caja, con los datos ingresados en la entrada “x”. Genera y devuelve el ID de la apertura con el que se inserta el elemento.

public **JSONObject** Venta(**JSONObject** x):

Inserta una nueva entrada a la tabla Venta, para esto comprueba si existen suficientes productos en la sucursal donde se realiza la compra para satisfacer la demanda y devuelve el mensaje de que no hay suficiente. Luego para cada elemento de contenido en la entrada “x” agrega un contenido a la tabla cont_venta.

public int RebajarCantidadVenta(**int** cantidad, **int** sucursal, **string** idmedicamento):

Resta de la cantidad total en la sucursal (tabla suc_medicamento) y la cantidad global para esa compañía de ese medicamento (tabla medicamento):

public class Cliente:

public **JSONObject** TodosClientes():

Este método devuelve todos los clientes registrados almacenados en el sistema.

public **JSONObject** ObtenerCliente(**int** cedula):

Este método realiza una devolución de la información general de un cliente obteniéndolo por el número de cedula.

`public JObject InsertarCliente(JObject x):`

Recibe en la entrada "x", los datos del Cliente a insertar, y procede a insertarlo en ambas bases de datos.

`public JObject ActualizarCliente(JObject x):`

Recibe los datos de un cliente y modifica los campos correspondientes en el cliente que corresponda al número de cedula del paquete de datos "x".

`public JObject EliminarCliente(int cedula):`

Realiza un borrado lógico del cliente al que corresponda el número de cedula ingresado, esto cambiando el valor del atributo "Activo" a 0.

`public class Empleado:`

`public JObject TodosEmpleados(int idcomp):`

Devuelve todos los empleados de una compañía específica.

`public JObject Login(int ced, string pass):`

Este método comprueba que exista en la base de datos un empleado con numero de cedula (ced) y una contraseña (pass).

Si el login es incorrecto manda un aviso. Si es correcto revisa cual es el rol del empleado.

Si el empleado es administrador: devuelve la información de la sucursal que administra y la información del empleado.

Si es un cajero: devuelve la información del Empleado y el id de la sucursal en la que está.

`public JObject ObtenerEmpleado(int ced):`

Este método devuelve la información de un empleado según la cedula ingresada (ced).

`public JObject EliminarEmpleado(int cedula):`

Realiza un borrado lógico de un empleado según la cedula que se ingresa (ced), una eliminación lógica consiste en cambiar el atributo activo a 0.

`public JObject ActualizarEmpleado(JObject x):`

Recibe en la entrada "x", todos los datos del empleado, y actualiza los campos en la base de datos para el empleado cuyo número de cedula sea igual.

`public JObject InsertarEmpleado(JObject x):`

Inserta un empleado en la base de datos, devuelve un error si ya existe un empleado con el mismo número de cedula.

`public class Medicamento:`

`public JObject TodosMedicamentosSucursal(int id):`

Devuelve la información de los medicamentos que están en la sucursal de id igual al ingresado (id).

`public JObject TodosMedicamentosComp(int id):`

Devuelve la información de los medicamentos que están relacionados al id de compañía ingresado (id).

`public JObject InsertarMedicamentoComp(JObject x):`

Inserta en la base de datos un nuevo medicamento con atributos iguales a los que se ingresan para la compañía ingresada.

`public JObject InsertarMedicamentoSuc(JObject x):`

Inserta en la table suc_medicamento un medicamento relacionado a una sucursal, dejando su cantidad como la ingresada y aumentando el total den la tabla medicamento en la cantidad que se ingresa a la sucursal.

`public JObject EliminarMedicamentoSucursal(string idmedicamento):`

Realiza una eliminación lógica de un elemento en la tabla med_suc y reduce la cantidad total en la tabla de medicamento del medicamento de dicha tabla para la compañía que pertenece la sucursal.

`public JSONObject EliminarMedicamento(string idmedicamento):`

Elimina un medicamento general de una compañía y elimina sus existencias en la tabla suc_medicamento, donde las sucursales sean de la misma compañía.

`public JSONObject ActualizarMedicamentoSuc(JSONObject x):`

Actualiza la información de un medicamento, con la información ingresada en la entrada "x", buscándolo por el id de medicamento y el id de sucursal contenidos en dicha entrada.

`public JSONObject ActualizarMedicamentoComp(JSONObject x):`

Actualiza los datos de un medicamento con la información ingresada en la entrada "x", buscándolo por el id del medicamento y la compañía a la que pertenece contenidos en dicha entrada.

`public class Sucursal:`

`public JSONObject TodosSucursales(int idcomp):`

Obtiene la información relevante de todas las sucursales de la compañía ingresada (idcomp).

`public JSONObject ObtenerSucursal(JSONObject x):`

Obtiene una sucursal buscándola por su nombre y la compañía a la que pertenece.

`public JSONObject InsertarSucursal(JSONObject x):`

Inserta una sucursal con los datos ingresados en "x" y coloca como administrador al empleado por defecto de cedula "0".

`public JSONObject ActualizarSucursal(JSONObject x):`

Actualiza los datos de una sucursal con los datos ingresados en “x”, con el id de sucursal que este trae.

`public JObject EliminarSucursal(int id):`

Realiza una eliminación lógica de una sucursal y elimina lógicamente las entradas en suc_medicamento que poseen el id de la sucursal eliminada.

`public class Proveedor`

`public JObject ObtenerProveedores():`

Obtiene todos los proveedores guardados en el sistema.

`public JObject InsertarProveedor(string nombre):`

Inserta en la table proveedores un nuevo proveedor con el nombre ingresado y una llave autogenerada.

`public JObject ActualizarProveedor(int id):`

Actualiza el nombre del proveedor con el id ingresado (id).

`public JObject EliminarProveedor(int id):`

Elimina el proveedor según el id ingresado (id).

4. Descripción detallada de las Estructuras de datos Utilizadas.

JSON:

Acronimo de “Java Script Object Notation”. Es una estructura utilizada para el manejo de datos, común mente utilizada en el desarrollo de comunicaciones web.

Ejemplo:

```
var jason = {  
  "age" : "24",  
  "hometown" : "Missoula, MT",  
  "gender" : "male"  
};
```

JSONObject:

Es una estructura utilizada para manejar el tipo de dato JSON desde CSS.

Ejemplo: {

```
string json = @"{
    CPU: 'Intel',
    Drives: [
        'DVD read/writer',
        '500 gigabyte hard drive'
    ]
}";

JSONObject o = JSONObject.Parse(json);

}
```

JArray:

Consiste en un arreglo que contiene estructuras JSON dentro de él, como cada elemento.

Ejemplo:

```
string json = @"[
    'Small',
    'Medium',
    'Large'
]";

JArray a = JArray.Parse(json);
```

5. Problemas Conocidos.

- Al inicio del proyecto para la conexión entre Postgresql y el API se optó por utilizar un framework llamado Devart dotConnect for PostgreSQL el cual según se investigó se encargaba de conectar visual studio con la base de datos en Postgresql. Pero a la hora de utilizarla para realizar la conexión dio problemas que impedían la misma. El cual no se pudo arreglar por esto se optó por utilizar otro método de conexión.
- A la hora de realizar una función que retorne valores, este no se retornaba en cambio tira un mensaje de error el cual menciona que no se pueden retornar datos.
- No se pudo subir el servicio de “Reporting Services” a la nube de Microsoft Azure.

6. Problemas Encontrados.

- Se tuvo un problema para la realización de la conexión entre Postgresql y el API mediante la creación del Entity Framework. Inicialmente se utilizó Devart dotConnect for PostgreSQL esta aplicación causaba un error y no permitía la conexión con Postgresql. El error de conexión se solucionó utilizando Npgsql para la conexión con la base.
- A la hora de conectar la base de datos en Postgresql con Crystal Report se tuvieron varios problemas con la conexión. Inicialmente se estaba utilizando una versión del paquete Npgsql incompatible con la versión de Visual Studio utilizado, por lo cual se procedió a utilizar la versión correcta lo cual se corrigió el error, pero igualmente no se logró la conexión con la Azure.
- A la hora de realizar la conexión de Crystal Report con Azure los DNS daban un error el cual mencionaba que no se podía realizar la conexión ya que el DNS ingresado no era válido para la conexión. Este error consistía en que la aplicación web que se está realizando para este segundo proyecto está en un formato de 32 bits y la ODBC de PostgreSQL instalado se encontraba en un formato de 64 bits lo cual creaba conflicto en los datos. El error se corrigió instalando los drivers de 32 bits del ODBC de PostgreSQL, esto corrigió el error.
- Nuevamente hablando de la conexión de Azure con Crystal Report. Se tuvo un problema nuevamente con los DNS en este caso las conexiones creadas anteriormente se realizaron con una arquitectura de 64 bits, esto causaba que Crystal Report no encontrara el DNS ya que se encontraba en distintas rutas. Se cambiaron las rutas de los DNS a sus rutas respectivas en 32 bits lo cual soluciono el error y finalmente se logró la conexión entre Crystal Report y la Base en Azure.
- Overflow del tipo numeric: La precisión y la escala del valor utilizado no eran lo suficiente para poder almacenar el resultado total, pues el valor de la precisión no fue hecho tomando en cuenta que la escala va incluida en este por ende la cantidad de dígitos para la parte entera del valor era muy pequeña.
- A la hora de hacer de enviar una petición de Delete al API, C# no permite enviar los datos como JSON en el cuerpo del request del mensaje. Para lograr hacer un delete lógico en la base de datos, los atributos necesarios deben ser enviados en el cuerpo del URL.

7. Recomendaciones y Conclusiones.

- Recomendaciones.

- En base al uso de Entity Framework se recomienda esperar tener la base es un punto en el cual completa (se espere no realizarle más cambio) antes de jalar los modelos y funciones producidos por el Entity ya que cada vez que se actualiza la base se debe volver a crear el Entity.
- Se recomienda para la conexión entre PostgreSQL y Visual Studio utilizar Npgsql para realizar la referencia con el Entity ya que este facilita mucho su utilización con PostgreSQL.
- La utilización de Entity es de mucha ayuda ya que es una herramienta que permití simplificar el proceso de creación de funciones y la conexión con la base de datos. Pero este al ser un Framework, no es una herramienta tan libre o flexible como sqlConnector (La función utilizada en el primer proyecto para realizar la conexión con SQL SERVER). Por lo cual se recomienda leer bastante sobre la utilización de Entity Framework antes de su implementación para conocer mas las ventajas y las limitaciones que presenta.
- Se recomienda que a la hora de utilizar Crystal Report para la creación de reportes, hay que tener en cuenta las arquitecturas del proyecto y del sistema ya que esto se toma en cuenta a la hora de realizar la conexión con PostgreSQL, al igual hay que tener cuidado con las rutas utilizadas.
- Se recomienda utilizar una conexión normal no utilizando el Entity Framework para llamar las funciones de POSTGRESQL, pues el Entity Framework es poco amigable para el uso de estas.

- Conclusiones.

- Crystal Report es una aplicación que facilita la creación de Reportes de los datos pertenecientes a una base de datos. Ya que esta permite obtener los datos directamente de la base de datos. Pero pese a esto hay que tener cuidado con su uso, ya que es muy tosco y no muy flexible. Pero la creación de reportes es una buena herramienta.
- La utilización de Entity Framework aligerar la carga de la conexión entre Visual Studio y la base de datos en este caso PostgreSQL, ya que la misma se encarga de crear las tablas en forma de clases y sus funciones necesarias para su fácil utilización. Por lo cual es una herramienta muy útil a la hora de crear aplicaciones web.
- POSTGRESQL es una excelente alternativa a SQL, y posee una gran cantidad de herramientas que permiten flexibilidad a la hora de desarrollar una base de datos.
- POSTGRESQL es ampliamente soportado por los diferentes servicios de gestión web de bases de datos y diferentes aplicaciones online.

8. Bibliografía.

- The PostgreSQL Global Development Group, T. (2017). *PostgreSQL: Documentation: 8.1: Modifying Tables*. [online] Postgresql.org. Available at: <https://www.postgresql.org/docs/8.1/static/ddl-alter.html> [Accessed 15 Nov. 2017]
- jrummell, j. (2017). *Entity Framework - stored procedure return value*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/10339750/entity-framework-stored-procedure-return-value> [Accessed 10 Nov. 2017].
- The PostgreSQL Global Development Group, T. (2017). *PostgreSQL: Documentation: 9.3: Numeric Types*. [online] Postgresql.org. Available at: <https://www.postgresql.org/docs/9.3/static/datatype-numeric.html> [Accessed 12 Nov. 2017].
- W3schools.com. (2017). *W3Schools Online Web Tutorials*. [online] Available at: <https://www.w3schools.com/> [Accessed 5 Nov. 2017].
- zszep, z. (2017). *How to display a server side generated PDF stream in javascript sent via HttpResponseMessage Content*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/21950828/how-to-display-a-server-side-generated-pdf-stream-in-javascript-sent-via-httpmes> [Accessed 10 Nov. 2017].
- SelectPDF. (2017). *HTML To PDF API*. [online] Available at: https://selectpdf.com/html-to-pdf-api/?gclid=Cj0KCQiA3dTQBRDnARIsAGKSflm5lrQIHIOyEFLdQteBQwx5J3d96rzv1ioj1qw0SatTKiaRipL3gVlaAuHiEALw_wcB [Accessed 10 Nov. 2017].
- C-sharpcorner.com. (2017). *Export Crystal Report in Various Formats Using ASP.Net*. [online] Available at: <http://www.c-sharpcorner.com/UploadFile/b926a6/export-crystal-report-into-pdfword-and-excel-file-formats-u/> [Accessed 9 Nov. 2017].
- Reitan, E. (2017). *Creating a Basic ASP.NET 4.5 Web Forms Page in Visual Studio 2013*. [online] Docs.microsoft.com. Available at: <https://docs.microsoft.com/en-us/aspnet/web-forms/overview/getting-started/creating-a-basic-web-forms-page> [Accessed 10 Nov. 2017].
- Docs.microsoft.com. (2017). *Connect to Azure Database for PostgreSQL from C#*. [online] Available at: <https://docs.microsoft.com/en-us/azure/postgresql/connect-csharp> [Accessed 13 Nov. 2017].
- RojanskyS, R. (2017). *Npgsql PostgreSQL Integration - Visual Studio Marketplace*. [online] Marketplace.visualstudio.com. Available at: <https://marketplace.visualstudio.com/items?itemName=RojanskyS.NpgsqlIPostgreSQLIntegration> [Accessed 13 Nov. 2017].

- Rocshy, R. (2017). *Insert data using Entity Framework model*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/8835434/insert-data-using-entity-framework-model> [Accessed 10 Nov. 2017].
- Jaworski, P. (2017). *Entity Framework to json - grouping data*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/30429455/entity-framework-to-json-grouping-data> [Accessed 10 Nov. 2017].
- The PostgreSQL Global Development Group, T. (2017). *PostgreSQL: Documentation: 9.1: CREATE FUNCTION*. [online] Postgresql.org. Available at: <https://www.postgresql.org/docs/9.1/static/sql-createfunction.html> [Accessed 12 Nov. 2017].
- Tutorials Point, T. (2017). *PostgreSQL - AUTO INCREMENT*. [online] www.tutorialspoint.com. Available at: https://www.tutorialspoint.com/postgresql/postgresql_using_autoincrement.htm [Accessed 11 Nov. 2017].
- Piyush, P. (2017). *How to iterate JSON array in JavaScript?*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/15496508/how-to-iterate-json-array-in-javascript> [Accessed 13 Nov. 2017].
- user708683, u. (2017). *Angular : Have multiple functions in one Service*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/36655564/angular-have-multiple-functions-in-one-service> [Accessed 16 Nov. 2017].
- bluish, b. (2017). *Add new attribute (element) to JSON object using JavaScript*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/736590/add-new-attribute-element-to-json-object-using-javascript> [Accessed 17 Nov. 2017].
- Kenedy, J. (2017). *Angular js - Error: \$parse:lexerr Lexer Error*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/23093660/angular-js-error-parselexerr-lexer-error> [Accessed 17 Nov. 2017].
- insertusernamehere, i. (2017). *How to iterate JSON array in JavaScript?*. [online] Stackoverflow.com. Available at: <https://stackoverflow.com/questions/15496508/how-to-iterate-json-array-in-javascript> [Accessed 23 Nov. 2017].