

TECNOLÓGICO DE COSTA RICA

ÁREA DE INGENIERÍA EN COMPUTADORES

CE3101 – Bases de Datos

Proyecto 2:

TABAS

REALIZADO POR:

Campos Abarca Alejandro 2015035779

López Salas, Luis 2015088115

Murillo Morgan, Francisco J. 2015147156

PROFESOR:

Marco Rivera

I SEMESTRE DEL 2019 - GRUPO 2

Tabla de Contenidos

Tablas, 5

Stored Procedures Trigguers y Funciones, 6

Problemas Encontrados, 8

Modelo Relacional, 4

Modelo de Chen, 3

Descripción detallada de la arquitectura desarrollada, 7

Conclusiones, 8

Bibliografía:, 8

# Modelode Chen

# Modelo Relacional

# Tablas

|  |  |
| --- | --- |
| Roles | |
| PK | idRol int numero identificación de rol. |
|  | rol varchar(13) nombre del rol. |

|  |  |
| --- | --- |
| Trabajadores | |
| PK | username varchar(60) nombre de usurario |
|  | pw bytea contraseña |
|  | Nombre varchar(60) |
|  | apellido1 varchar(60) |
|  | apellido2 varchar(60) |
|  | cedula int |
| FK | rol int |

|  |  |
| --- | --- |
| Marcas | |
| PK | idMarca int identificador de la marca |
|  | Marca varchar(40) nombre de la marca |

|  |  |
| --- | --- |
| BagCarts | |
| PK | idBagCart int identificador de BagCart |
| FK | marca int marca del BagCart |
|  | modelo int año del modelo |
|  | Sello varchar(10) sello de seguridad |

|  |  |
| --- | --- |
| Estados | |
| PK | idEstado int Identificador del Estado |
|  | Estado varchar(60) Estado de la maleta |

|  |  |
| --- | --- |
| Maletas | |
| PK | idMaletas int Identificador de Maleta |
|  | username varchar(60) nombre del usurario |
|  | color varchar(15) color de la maleta |
|  | peso int peso de la maleta |
|  | costo int costo por maleta |

|  |  |
| --- | --- |
| MaletasBagCarts | |
| PK-FK | idMaleta int Identificador de Maleta |
| PK-FK | username varchar(60) nombre del usuario |
| FK | trabajadorRX varchar(60) encargado de rayos X a la maleta |
| FK | bagCart int Carrito que lleva maletas |
| FK | estado int Estado de la maleta |
|  | comentario text |

|  |  |
| --- | --- |
| Modelos | |
| PK | idmodelo int Identificador del Modelo |
|  | modelo varchar(60) Modelo del Avion |
|  | cantidadBodegas int Cantidad de Bodegas en el Avion |
| Aviones | |
| PK | idAvion int Identificador del Avion |
| FK | Modelo int Modelo del Avion |

|  |  |
| --- | --- |
| Bodegas | |
| PK – FK | avion int Avion al que pertenece |
| PK | idBodega int Identificador de la Bodega |
|  | capacidad int Cantidad de espacio para llevar maletas |

|  |  |
| --- | --- |
| MaletasAviones | |
| PK – FK | idMaleta int Identificador de Maletas |
| PK – FK | username varchar(60) Nombre de Usuario |
| FK | avion int Avion al que pertenece |
| FK | bodega int Numero de Bodegas |
| FK | trabajadorSc varchar(60) Encargado de escanear la maleta |

|  |  |
| --- | --- |
| Vuelos | |
| PK | id int Identificador de Vuelo |
| FK | avion int Avion al que pertenece |
|  | fecha timestamp Tiempo del vuelo |
|  | abierto boolean Estado del vuelo |
| FK | bagCart int Carrito que lleva maletas |

Stored Procedures Trigguers y Funciones

En el proyecto se utilizaron los llamados Stored Procedures, los cuales se encargan de que la base de datos haga acciones de una manera reutilizable. Se puede comparar a una función en un lenguaje de programación, se puede llamar en vez de crearlos cada vez que se utilicen. Los siguientes son los Stored Procedures usados en la base:

* procRegistrarTrabajador: Se encarga de crear un trabajador en la base de datos, recibe como parámetro todos los datos que contiene los trabajadores en la tabla Trabajadores, hace todas las validaciones necesarias para crear un usuario valido conforme a las restricciones de la base.
* procGetPasswordTrabajador: Se encarga de sacar la contraseña de la base encriptada, desencripta.
* procGetTrabajador: Desencripta los datos del trabajador para su uso.
* procGetCantidadMaximaDeBodegasPorAvion:
* trgg\_sello: Genera un sello alfanumerico para los bagcarts asignados a vuelos.
* procInsertBodega: Revisa la entidad Modelos y valida que un determinado avión no pueda tener más bodegas de las permitidas por su modelo.
* procCerrarVuelo: Se encarga de cerrar un vuelo.
* trgg\_quitarSello: Quita el sello del BagCart cuando el vuelo es cerrado.
* proc\_asignarBagCartVuelo: Valida la actualización de Vuelos a la hora de asignar un bagcart.
* proc\_asignarMaletaBagCart: Asigna un bagcart a la maleta.
* proc\_asignarMaletaAvion: Asigna un BagCart al avión.
* random\_string: Genera un string random de una longitude especificada como parámetro.

Descripción detallada de la arquitectura desarrollada

Para este segundo proyecto, los estudiantes realizaron dos páginas web, dos bases de datos, y un solo servidor rest para la comunicación entre las paginas y las bases de datos. Primero se realizará una descripción acerca de las bases de datos utilizadas. La primera de estas y la principal de este proyecto será tabas. Tabas estará alojada en el servidor de azure. Esta plataforma nos permitirá acceder a los datos de la base desde cualquier conexión remota que halla sido especificada previamente. Aquí es donde entra el NancyServer. El NancyServer es un webService el cual accederá a la base de datos y retornará datos específicos según la dirección que se utilice. El server puede ser accesado a través de la siguiente url : <https://nancyrest.azurewebsites.net/>. Por último, estarán las dos paginas web. La primera será la página de administración, esta página será la encargada de registrar usuarios y trabajadores a la plataforma. También será encargada de asignar bagcarts, crear bagcarts, y crear maletas. Para la segunda plataforma, tendrán acceso los scanner y embarcadores, los cuales tendrán acceso a asignar maletas a los bagcart, rechazar maletas, y asignar las maletas en un bagcart a una bodega de un avión.

Problemas Conocidos

* A la hora de iniciar sesión no envia el aviso de error pero este no le permite acceder a ninguna otra parte de la página.
* El lector qr fue incompatible con Angular por lo tanto no se implemento.

Problemas Encontrados

* Postrges SQL aunque es una plataforma útil, al tener sutiles diferencias con SQL Server, se complicó realizar procedimientos y trigguers ya que estos cambian muy drásticamente.
* Azure no permite la creación de funciones determinadas por los usuarios en SQL, por lo tanto esto impidió poder montar la base de datos TECAirlines en la plataforma.
* El scanner de códigoQR para web es bastante inestable, por lo que presentaba problemas con angular por lo que se optó por no utilizarlo.

Conclusiones

A la hora de realizar este proyecto se adquirieron conocimientos acerca de servicios web. Se lográ concluir que es más sencillo desplegar el servidor REST asi como la base de datos en un servicio de estos, ya que simplifica la conexión con las plataformas que deben de acceder a esta información. Además de esto se logra concluir que existen diversos motores de bases de datos, todos con sus ventajas y desventajas, y queda a decisión del programador cual motor utilizar.

Bibliografía:

[3] Angular 6 - User Registration and Login Example & Tutorial. (2018, May 16). Retrieved March 1, 2019, from <http://jasonwatmore.com/post/2018/05/16/angular-6-user-registration-and-login-example-tutorial>

GitHub: https://github.com/fmurillom/CE-3101BD-TABAS