

IMPLEMENTACION DE SISTEMAS

ANÁLISIS PARA LA IMPLEMENTACIÓN DE UN PROTOTIPO DE RASTREO DE TRANSPORTE URBAN TAMPICO



Integrantes:

- Hugo Enrique Cecenes Morales
- Jesús Alexander Gomez Enríquez
- Claudia Paola Castillo Abundis
- Abel Rivera Valdez

Prof. Cynthia Gladys

INGENIERÍA EN SISTEMAS
COMPUTACIONALES Y ELECTRÓNICOS

Universidad del Noreste

Tabla de contenido

INTRODUCCIÓN.....	2
ANTECEDENTES.....	3
DEFINICIÓN DEL PROBLEMA.....	7
JUSTIFICACIÓN.....	7
OBJETIVO GENERAL.....	7
OBJETIVOS ESPECÍFICOS.....	8
CRONOGRAMA.....	8
ENTREGABLES.....	9
DIAGRAMA ENTIDAD-RELACION.....	9
DESCRIPCION DE LAS TABLAS Y SUS CAMPOS.....	9
Tabla Usuarios.....	9
Tabla Ruta.....	10
Tabla Autobús.....	10
Tabla UrbanApp.....	10
ESQUEMA DE LA BASE DE DATOS.....	11
SCRIPTS DE LA CREACION DE LA BASE DE DATOS.....	11
MODULOS DEL SISTEMA.....	14
Conexión.php.....	14
Registro.php.....	15
Login.php.....	15
Logout.php.....	16
CASOS DE PRUEBA.....	16
IMPLEMENTACION.....	18
MANUALES DE USUARIO.....	18
MANUAL TECNICOS.....	18

INTRODUCCIÓN

El transporte urbano siempre ha sido de mucha utilidad en cualquier ciudad, debido a que personas que no cuentan con auto propio necesitan trasladarse ya sea para ir a su trabajo o por otros motivos.

La capacidad del transporte depende del tamaño de la ciudad (población) y por ende el número de unidades.

Además, las empresas de transporte adquieren unidades diferentes en base a la demanda de pasajeros transportados o también en la distancia recorrida ya que algunas son rutas largas o cortas.

El prototipo web de quejas y denuncias representa una herramienta para el control y mejoramiento continuo de la empresa o institución, toda vez que posibilita el conocimiento sobre la percepción de los clientes de los bienes o servicios que se comercializan; lo cual se complementa con una automatización informática mediante un Sistema electrónico en un portal Web, donde las inquietudes, quejas y sugerencias que tienen los usuarios con respecto a los servicios, serán reportadas en ellos con miras a mejorar la calidad de los mismos. Para describir un sistema de quejas y denuncias, es importante conocer cada uno de los conceptos involucrados.

El Diccionario de la lengua española, nos proporciona una explicación de lo que es un sistema:

“Conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí.”

Una queja es la inconformidad del cliente o de la ciudadanía en la contraprestación de un bien, o en la indebida prestación de un servicio público o privado, o también la inconformidad en el trato recibido por sus funcionarios; por decirlo de otra forma, es el mal funcionamiento de las instituciones privadas o públicas, por la mala calidad de sus productos, servicios o el trato inadecuado de sus empleados. Sin embargo, los sistemas convencionales en algunos casos son ineficientes y no redituables, en concepto de empleado, materia prima etc. Al implementar el SQD le facilitamos tanto a los usuarios realizar la queja y a la empresa el recibirlas.

ANTECEDENTES



¿Qué es Uber? Con Uber un GPS localiza tu ubicación, luego eliges el automóvil que prefieres, ves el perfil y la foto del chofer, lo contratas con tu celular. En unos cuantos minutos aparece ante ti algún modelo de clase élite listo para llevarte a tu destino con una atención

Premium.

¿Dónde surgió? Uber es un **servicio de taxis de lujo que inició en San Francisco California** y que hoy opera en más de 200 ciudades en todo el mundo a través de una “app” que se instala y se maneja muy fácil en Smartphones.

Uber en México: Uber es una empresa legalmente constituida en México y tiene choferes registrados como contribuyentes que pagan sus impuestos, opera en el **Distrito Federal, Guadalajara, Tijuana y Monterrey.**

Pasos para tener el servicio Uber:

- Al solicitar un vehículo Uber, la aplicación vía GPS identifica nuestra ubicación (a donde queremos que pasen por nosotros).
- Después, debemos introducir nuestro destino y seleccionar el tipo de servicio Uber que queremos (básico, camioneta, de lujo, etc.). Si deseamos, podemos calcular la tarifa aproximada del viaje.
- Al solicitar el servicio de Uber podemos observar los carros que están próximos a nosotros y después de unos segundos, **se nos notifica qué carro vendrá por nosotros**, el nombre del chofer y su foto.
- Mientras se espera el servicio, la aplicación tiene algunas opciones como contactar al chofer a su celular (por ejemplo para dar alguna indicación extra) o bien cancelar, si fuera necesario. **Según el chofer, cuando llega al lugar donde pedimos el servicio, puede lanzar una alerta por medio de la aplicación o bien hacer una llamada para avisar que ha llegado.**
- Una vez que el vehículo Uber llega por nosotros, el chofer nos abre la puerta y nos ofrece agua embotellada. El celular del chofer ya tiene nuestra dirección de destino, así como una ruta que podemos aceptar o cambiar.



- Otro dato es que como Uber tiene dada de alta una tarjeta de crédito nuestra, al llegar al destino no se paga en efectivo. Cuando termina el viaje, el chofer cierra el servicio e inmediatamente llega a nuestro correo electrónico el “Recibo Uber ”que indica cosas como: fecha, ubicación de inicio y ubicación final del viaje, horario de inicio y final del viaje, kilómetros recorridos, desglose de la tarifa, foto y nombre del chofer.



Life360
The new family circle

¿En qué consiste Life360?
Actualmente una de las principales preocupaciones de los padres es saber en dónde se encuentran sus hijos cuando salen de casa, por eso se creó Life360,

aplicación para dispositivos móviles, ya sean Android o iOS, que permite tener conectada a la familia y saber en dónde se encuentran en todo momento. Life360 ofrece el servicio de localización familiar más preciso, gracias a su moderna tecnología de GPS.

Con esta aplicación podrás:

- Ver la ubicación de tu familia en un mapa accesible solo por invitación: Permite ver en donde se encuentran todos los integrantes de tu familia en un mapa privado que se basa en la tecnología GPS, WiFi y de triangulación (para teléfonos que no son Smartphone).
- Elegir cuándo compartir la ubicación: Esta aplicación también permite elegir cuando quieres compartir tu ubicación y desactivarla cuando no quieras que te localicen.
- Chatear de forma individual o en grupos: Life360 permite chatear con una persona en específico o de manera grupal, puedes organizar reuniones familiares y saber en qué momento llegan, o en que parte del camino se encuentran.
- Recibir alertas cuando los miembros llegan a lugares favoritos: Puedes guardar los lugares que visitas con más frecuencia y recibir alertas automáticas cuando algún miembro de la familia llegue o se vaya.
- Rastrear un teléfono perdido o robado: Permite localizar un teléfono que haya sido robado, solo basta con entrar en el teléfono de algún integrante de la familia y ver en donde se encuentra tu celular.
- Botón del pánico: Esta aplicación



cuenta con el recurso del botón del pánico, con el cual permite hacer una llamada, enviar un email o un mensaje de texto a los miembros de la familia indicando las coordenadas de GPS en las que te encuentras en ese momento. Un beneficio de esto, es que cuando las líneas telefónicas fallan, tú puedes comunicarte por medio de esta opción.



Moovit es la aplicación líder mundial para el transporte público que cada día utilizan más de 28 millones de personas en más de 750 países. Gracias a la información más exacta y precisa que cualquier otra

aplicación de movilidad, Moovit incorpora una nueva ciudad cada 24 horas. Combinando todas las opciones de transporte público en una única app, Moovit te permite obtener el control total de tus trayectos en transporte público, disfrutando de una tranquilidad hasta ahora desconocías.

Debido a que el transporte público es siempre impredecible, Moovit realiza actualizaciones constantes ante cualquier cambio o alteración del servicio de manera que no viajarás a una parada que está cerrada o no esperarás un autobús que no está funcionando.

- **Vamos y Notificaciones** te guían paso a paso en tiempo real durante tu trayecto. Sabrás exactamente hacia donde debes caminar, cuánto durará la espera y cuántas paradas quedan hasta tu destino. Así podrás centrarte en cualquier otra cosa durante tu viaje ya que Moovit te avisará cuándo llegues a tu parada.
- **Bicicletas Compartidas** - Para los días soleados, días en los que no desees caminar, días en que te gustaría hacer un poco de ejercicio extra y muchos otros días.
- **Tiempo real de llegada** (en ciudades en las que esté disponible) te permite saber cuándo debes estar en una parada o cuánto tiempo tienes para hacer cualquier otra cosa mejor que esperar en la parada.
- **Cobertura global** en más de 750 ciudades lo que significa que, cuando viajes, podrás usar el transporte público como cualquier usuario local.

CARACTERÍSTICAS ADICIONALES:

- **La Pantalla de Favoritos** te permite guardar ubicaciones y líneas usadas frecuentemente con un solo toque.
- **El Widget** te da acceso inmediato a tus líneas y ubicaciones favoritas haciendo la comprobación de transporte tan fácil como revisar el estado del tiempo.
- **Alertas de Servicio** que te mantienen informado para que no te quedes esperando un autobús que no llegará o sentado en la estación de tren a

causa de un retraso. Al guardar tus líneas favoritas Moovit te enviará un mensaje cuando haya un problema que podría afectar tus viajes.

- La función Vamos viaja contigo y te alerta cuando te acercas a tu destino o cuando debas bajar. Así que aun cuando sabes exactamente a dónde vas, puedes darte el lujo de soñar despierto.
- Líneas y Mapas sin Conexión te permiten ver cualquier mapa de tránsito local disponible en un archivo PDF.

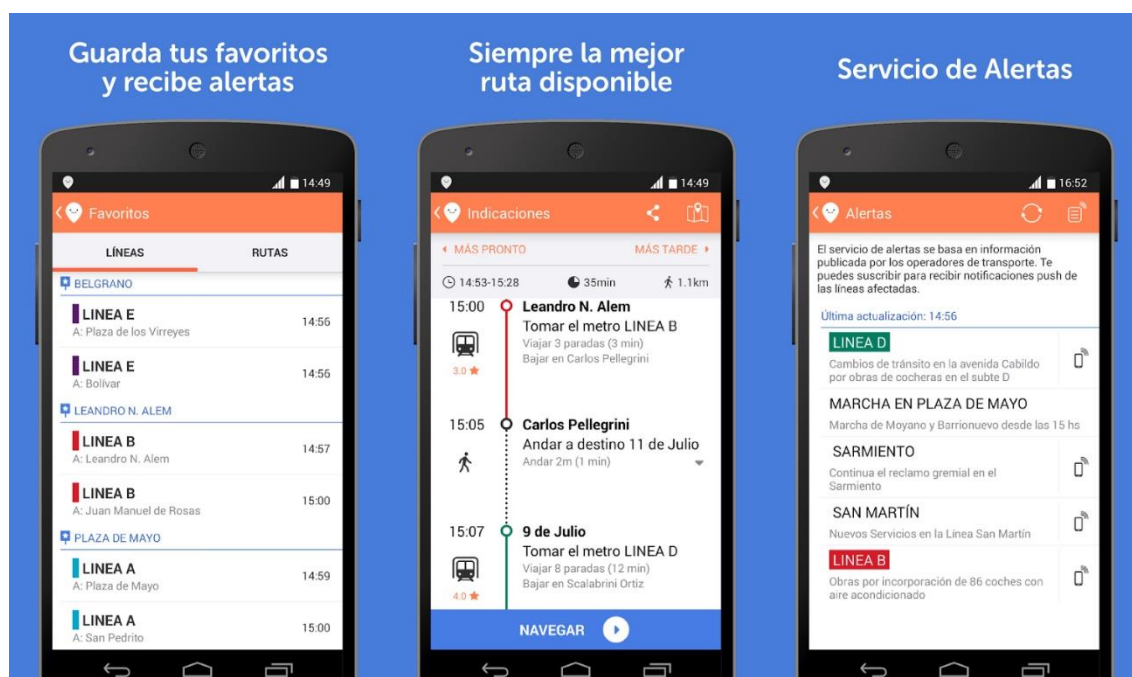
Usa Moovit en las principales ciudades de todo el país:

Buenos Aires, Córdoba, Rosario, Santiago, Concepción, Valparaíso y Viña del Mar, Bogotá, Cali, Medellín, Guadalajara, México DF, Monterrey, Montevideo y más...

Moovit colabora con los principales operadores de transporte a nivel nacional, tales como:

Metro de Santiago, Transantiago, Colectivos, Subte de Buenos Aires, Metrovias, Ferrovías, SITP, Transmilenio, MIO, Metro de Medellín, RTP, Metrobús, Pesero, MTOP y más...

¡Moovit es una aplicación disponible en todo el mundo! Puedes encontrarnos en cientos de ciudades de todo el mundo, tales como: Estados Unidos (Nueva York, San Francisco, Los Ángeles, Seattle), Reino Unido (Londres, Birmingham, Manchester), España (Madrid, Barcelona), Italia (Roma, Milán, Nápoles), Francia (París, Burdeos), Polonia (Varsovia, Lublin, y más), Suecia (Estocolmo), Finlandia (Helsinki), Hungría (Budapest), Rusia (Moscú), Países Bajos, Israel, Brasil (Río de Janeiro, Sao Paulo), Chile (Santiago), Colombia (Bogotá), Ciudad de México, Perú (Lima), Nueva Zelanda, Filipinas (Manila), Malasia (Kuala Lumpur), Grecia (Atenas), Corea del Sur (Seúl), Noruega (Oslo).



DEFINICIÓN DEL PROBLEMA

Los usuarios del transporte público “Urban Tampico” no suelen saber en cuanto tiempo se demorará dicho transporte. En ocasiones un autobús se demora por lo menos de 10 a 12 minutos y en otras ocasiones 2 o 3 autobuses de la misma empresa pasan uno tras otro. Las quejas y comentarios se manejan en el antiguo formato de llenado manual.

El principal problema en los sistemas de quejas radica en la atención/entrega de las mismas ya que los procesos de elaboración de estas cuentan con un sistema deficiente el cual depende de papelería, tramites, procesos complicados, etc. En ocasiones en todo ese proceso las quejas llegan a un estado determinado “Queja Muerta” donde la compañía pierde la queja y queda desatendida.

Los empleados son otro de los factores influyentes en este proceso ya que en ocasiones interfieren o simplemente desaparecen las quejas, esto para evitar reclamo debido a la inconformidad del cliente si es de una de las responsabilidades del mismo por lo cual se levanta la queja.

JUSTIFICACIÓN

La razón por la que se quiere desarrollar ésta aplicación es para ayudar a los pasajeros que utilizan el transporte Urban a tener información exacta (sea visual y escrita) sobre la red de rutas que existen.

Además de un enlace web con opción de quejas y comentarios sobre el servicio. La creación del prototipo web plantea que es posible resolver uno de los aspectos más comunes en la insatisfacción de un producto o servicio, la experiencia del consumidor al momento de presentar una queja a las compañías. esto con la intención de balancear el equilibrio que existe entre la oferta de calidad por parte de la empresa en cuestión y la satisfacción del cliente de forma en que se ven beneficiadas ambas partes; además de la automatización del sistema con el fin de agilizar y reducir costos en la recepción de las quejas para su oportuna resolución.

El prototipo como tal surge como la unión del proyecto urban y el proyecto sistema de quejas, al llegar a la conclusión que estos dos podrían resultar más eficientes al llevarse a cabo de manera conjunta ya que los dos van enfocados a la satisfacción del cliente y mejora de servicio, resultando esto benefico para ambas partes, (cliente-empresa)

OBJETIVO GENERAL

- Diseño e implementación de un prototipo móvil y web de rastreo de unidades correspondiente a la empresa Urban Tampico.

OBJETIVOS ESPECÍFICOS

- Hacer un conteo sobre cuántas unidades hay en servicio activo.
- Establecer costo o no al dispositivo de rastreo.
- Realizar estudio para saber cómo se va a recuperar la inversión.
- Detallar el funcionamiento del prototipo, qué ventajas tiene y en qué plataforma se encuentra disponible.
- Recabar datos sobre las rutas con las que cuentan, las frecuencias diarias de dichas rutas y los pasajeros que transportan diariamente.
- Las quejas sean transparentes para que los demás usuarios vean la fiabilidad del producto.
- Una queja deberá ser fácil de realizar para los usuarios.
- La automatización de las quejas permitirá que no exista tanto expediente, sino que todo sea tramitado por computadora y correo electrónico, para que la rapidez sea la cualidad de la resolución de las inconformidades.

CRONOGRAMA

Nombre	Fecha de inicio	Fecha de fin
Analisis para la Implementacion de un prototipo de rastreo de Transporta(URBAN Tampico)	16/05/16	25/07/16
creacion del prototipo web en asp.net	13/06/16	13/06/16
Investigacion de la instalacion de un servidor local	14/06/16	15/06/16
Instalacion del servidor local	15/06/16	15/06/16
Desarrollo de la base de datos de los usuarios	16/06/16	17/06/16
Desarrollo de pantallas de sistema de registro	17/06/16	17/06/16
Desarrollo de pantallas de acceso	17/06/16	17/06/16
Diseño de las pantallas esperadas	20/06/16	20/06/16
Desarrollo de la programacion en C# del registro de usuarios	21/06/16	22/06/16
Conexion de asp.net a sql sever	21/06/16	23/06/16
Rediseño de la programacion de sistema de registro	24/06/16	24/06/16
Rediseño de la base de datos	27/06/16	27/06/16
Desarrollo de los campos en los registros de usuarios	27/06/16	27/06/16
Desarrollo de la validacion de usuario	27/06/16	27/06/16
Creacion de prototipo UrbanMaps	28/06/16	28/06/16
Desarrollo de la programacion del trazado de rutas C#	29/06/16	29/06/16
Proceso de importacion de asp.net a html y php	29/06/16	1/07/16
proceso de importacion de la base de datos SQL Server a MySQL	29/06/16	1/07/16
Rediseño del sitio web	30/06/16	1/07/16
Implementacion de programacion CSS	1/07/16	1/07/16
Programacion de navegacion entre pantallas	4/07/16	4/07/16
Creacion de las paginas web	4/07/16	5/07/16
Implementacion de Google Maps Rute en html	5/07/16	6/07/16
Desarrollo del Trazado de rutas	6/07/16	7/07/16
Investigacion del sistema de registro y acceso en android	7/07/16	8/07/16
Desarrollo del sistema	8/07/16	8/07/16
Investigacion de como insertar mapas en android	8/07/16	8/07/16

Nombre	Fecha de inicio	Fecha de fin
investigacion de la creacion de rutas predeterminadas en android	11/07/16	11/07/16
presentacion del proyecto	28/07/16	28/07/16

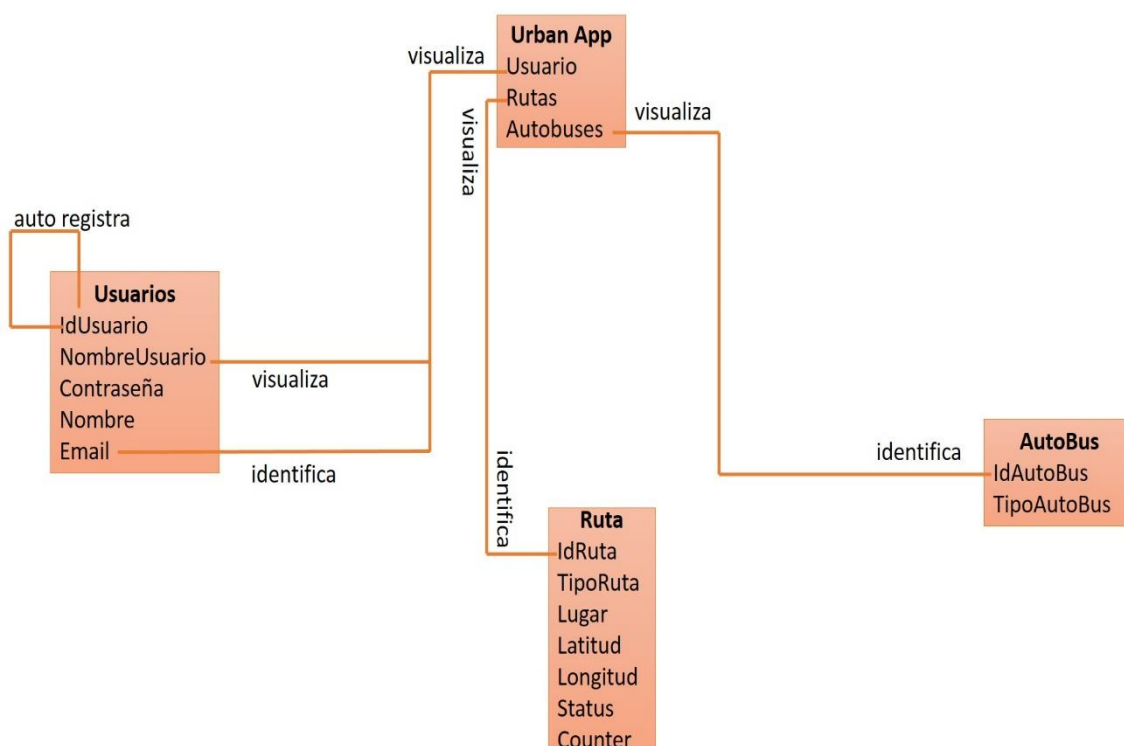
ARCHIVO ORIGINAL



cronograma.pdf

DIAGRAMA ENTIDAD-RELACION

El siguiente diagrama muestra el funcionamiento de la base de datos, cabe mencionar que la base de datos es usada tanto en la App como en la Web para que los usuarios puedan acceder a cualquiera de estos dos servicios.



DESCRIPCION DE LAS TABLAS Y SUS CAMPOS

La base de datos del proyecto cuenta con 4 tablas relacionadas entre sí, teniendo como tabla principal la de Usuarios, cuenta con dos tablas más llamadas Ruta y AutoBus y una cuarta tabla llamada UrbanApp la cual depende totalmente de las tres primeras. A continuación, se hace una breve y detallada descripción del contenido de estas tablas.

Tabla Usuarios. - Cuenta con 7 campos siendo el principal (llave primaria) el campo IdUsuario que es de tipo entero y no permite la entrada de datos nula este campo es el encargado de asignar al usuario un numero único de registro. El siguiente campo es NombreUsuario el cual es de tipo varchar con capacidad de hasta 50 caracteres, este campo contendrá el nombre con que el usuario registrado utilizara su cuenta, este campo no permite la entrada de datos nula; El campo Contraseña es de tipo varchar con capacidad de hasta 50 caracteres, este campo contendrá la clave con la cual el usuario se registrara en nuestra base de datos y la cual será requisito para poder ser reconocido por el sistema y determinar que se es el dueño de la cuenta creada, este campo no permite la entrada de datos nula; Los campos Nombre, ApPaterno y ApMaterno son de tipo varchar con capacidad de hasta 50 caracteres cada uno, estos contienen el nombre

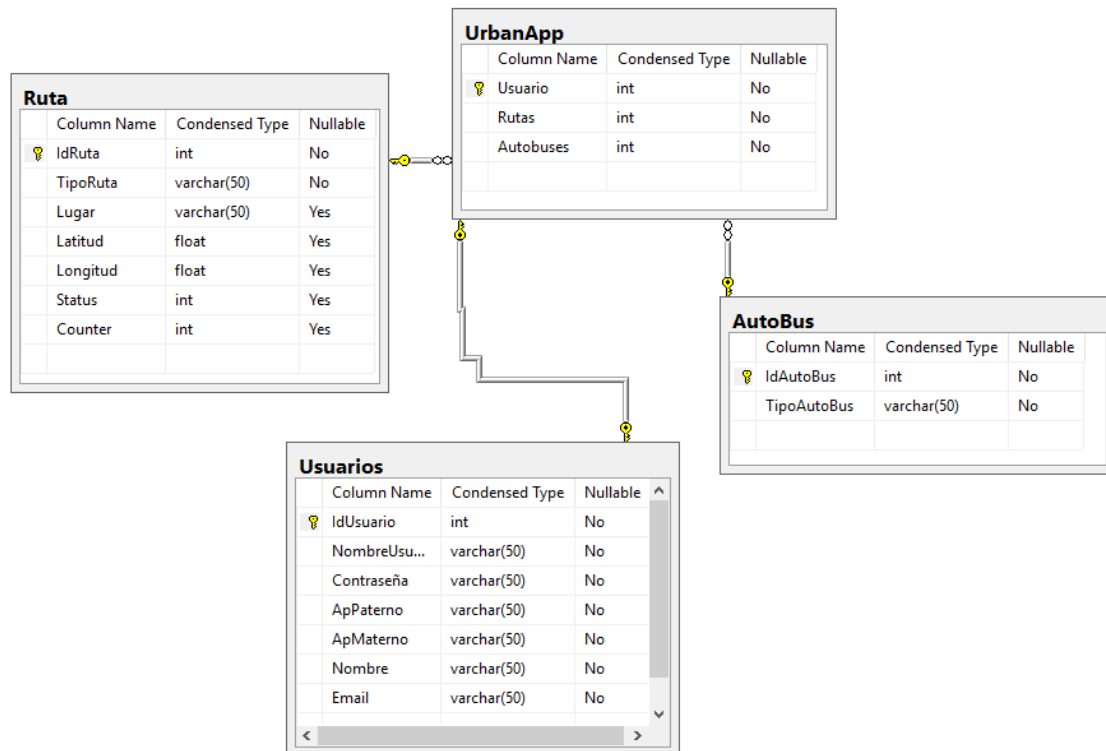
completo del usuario en conjunto y no permiten la entrada de datos nula; El campo Email es de tipo varchar con capacidad de hasta 50 caracteres este campo contiene la dirección de correo electrónico del usuario y no permite la entrada de datos nula.

Tabla Ruta. - Esta tabla cuenta con 7 campos siendo el principal el campo IdRuta de tipo entero este campo es el encargado de asignar a cada ruta un número de identificación diferente, no acepta valores nulos. El siguiente campo es el de TipoRuta de tipo varchar con capacidad de hasta 50 caracteres y que no acepta valores nulos, este campo será el encargado de contener el nombre de las rutas almacenadas; El campo Lugar es de tipo varchar con capacidad de hasta 50 caracteres, en este se marca una calle o parada en la cual pasara el autobús, este campo puede ser omitido ya que permite la entrada de valores nulos; Los campos Latitud y Longitud ambos de tipo float y que de igual forma que el anterior aceptan la entrada de valores nulos, estos campos contiene la información de las coordenadas sobre las que se mueve el vehiculo con la ruta previamente seleccionada al realizar la búsqueda; El campo Status de tipo entero de posible valor nulo, es el que contiene la información sobre si un autobús se encuentra retrasado o si este va a tiempo de acuerdo al horario de central; El campo Counter de tipo entero y de posible valor nulo, contiene la información del número de vueltas de la unidad seleccionada sobre la actual ruta.

Tabla Autobús.- Esta tabla cuenta con solo 2 campos, uno de ellos es el IdAutoBus el cual es de tipo entero y es la llave primaria, esto quiere decir que este campo contiene un numero con el que se reconocerá a cada autobús de modo que no se repita y no acepta valores nulos; El siguiente campo es TipoAutoBus que es de tipo varchar con capacidad de hasta 50 caracteres y que no acepta la entrada de valores nulos, este campo contiene la información acerca del tipo de autobús seleccionado.

Tabla UrbanApp.- Esta tabla contiene 3 campos, todos de tipo entero y que no aceptan la entrada de valores nulos, la característica principal de esta tabla es que se encuentra compuesta por la unión de las llaves primarias de nuestras 3 tablas anteriores estando la información de cada una en orden respectivo en cada uno de los campos que en ella se encuentra. A su vez cuenta con una llave primaria la cual consiste en la conexión entre la tabla de Usuarios utilizando el contenido del campo IdUsuario para alojarlo en este nuevo campo llamado Usuario; El siguiente campo es el que proviene del campo IdRuta y se encuentra en esta tabla como el campo Ruta que viene de la tabla Ruta; por ultimo tenemos el campo Autobuses el cual proviene de la llave primaria de la tabla AutoBus que es IdAutoBus.

ESQUEMA DE LA BASE DE DATOS



SCRIPTS DE LA CREACION DE LA BASE DE DATOS

```
USE [master]
GO

CREATE DATABASE [Aplicación]

GO

USE [Aplicación]
GO
/***** Object: Table [dbo].[Usuarios] *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Usuarios](
    [IdUsuario] [int] IDENTITY(1,1) NOT NULL,
    [NombreUsuario] [nvarchar](20) NOT NULL,
    [Contraseña] [nvarchar](20) NOT NULL,
    [ApPaterno] [nvarchar](20) NOT NULL,
    [ApMaterno] [nvarchar](20) NOT NULL,
    [Nombre] [nvarchar](20) NOT NULL,
    [Email] [nvarchar](30) NOT NULL,
    [CreatedDate] [datetime] NOT NULL,
    [LastLoginDate] [datetime] NULL,
    CONSTRAINT [PK_Usuarios] PRIMARY KEY CLUSTERED
)
```

```

        [IdUsuario] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO

USE [Aplicación]
GO

/***** Object: StoredProcedure [dbo].[Insertar_Usuario]*****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[Insertar_Usuario]
    @NombreUsuario NVARCHAR(20),
    @Contrasena NVARCHAR(20),
    @ApPaterno NVARCHAR(20),
    @ApMaterno NVARCHAR(20),
    @Nombre NVARCHAR(20),
    @Email NVARCHAR(30)
AS
BEGIN
    SET NOCOUNT ON;
    IF EXISTS(SELECT IdUsuario FROM Users WHERE Username = @NombreUsuario)
    BEGIN
        SELECT -1 -- Username exists.
    END
    ELSE IF EXISTS(SELECT IdUsuario FROM Users WHERE Email = @Email)
    BEGIN
        SELECT -2 -- Email exists.
    END
    ELSE
    BEGIN
        INSERT INTO [Usuarios]
            ([NombreUsuario]
            ,[Contraseña]
            ,[ApPaterno]
            ,[ApMaterno]
            ,[Nombre]
            ,[Email]
            ,[CreatedDate])
        VALUES
            (@NombreUsuario
            ,@Contrasena
            ,@ApPaterno
            ,@ApMaterno
            ,@Nombre
            ,@Email
            ,GETDATE())

        SELECT SCOPE_IDENTITY() -- UserId
    END
END

GO

/***** Object: Table [dbo].[ActivarUsuario]*****/
SET ANSI_NULLS ON

```

```

GO

SET QUOTED_IDENTIFIER ON
GO

go
CREATE TABLE [dbo].[ActivarUsuario](
    [IdUsuario] [int] NOT NULL,
    [ActivationCode] [uniqueidentifier] NOT NULL,
    CONSTRAINT [PK_ActivarUsuario] PRIMARY KEY CLUSTERED
(
    [IdUsuario] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

--Validar_usuario
GO
CREATE PROCEDURE [dbo].[Validar_usuario]
    @Email NVARCHAR(20),
    @Contrasena NVARCHAR(20)
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @IdUsuario INT, @LastLoginDate DATETIME

    SELECT @IdUsuario = IdUsuario, @LastLoginDate = LastLoginDate
    FROM Usuarios WHERE Email = @Email AND [Contraseña] = @Contrasena

    IF @IdUsuario IS NOT NULL
    BEGIN
        IF NOT EXISTS(SELECT IdUsuario FROM ActivarUsuario WHERE IdUsuario
= @IdUsuario)
        BEGIN
            UPDATE Usuarios
            SET LastLoginDate = GETDATE()
            WHERE IdUsuario = @IdUsuario
            SELECT @IdUsuario [IdUsuario] -- User Valid

        END
        ELSE
        BEGIN
            SELECT -2 -- User not activated.

        END
        END
        ELSE
        BEGIN
            SELECT -1 -- User invalid.

        END
        END
END

GO
INSERT INTO Usuarios
SELECT 'AlexGE', '12345', 'Gomez', 'Enriquez', 'Alexander', 'alex@gmial.com',
GETDATE(), NULL

SELECT * FROM Usuarios

create table AutoBus
(
    IdAutoBus int not null,
    TipoAutoBus varchar(50)not null,
    Constraint PK_AutoBus primary key (IdAutoBus)

```



```

)

create table Ruta
(
  IdRuta int not null,
  TipoRuta varchar(50) not null,
  Lugar varchar(50) null,
  Latitud float null,
  Longitud float null,
  Status int null,
  Counter int null,
  Constraint PK_Ruta primary key (IdRuta)
)

create table UrbanApp
(
  Usuario int not null,
  Rutas int not null,
  Autobuses int not null,
  Constraint PK_UrbanApp primary key (Usuario),
  Constraint FK_UrbanApp_Usuarios foreign key (Usuario) references Usuarios
(IdUsuario),
  Constraint FK_UrbanApp_Ruta foreign key (Rutas) references Ruta (IdRuta),
  Constraint FK_UrbanApp_Autobus foreign key (Autobuses) references Autobus
(IdAutoBus)
)

```

ARCHIVO ORIGINAL



DB-V1.2.sql

MODULOS DEL SISTEMA

Este código muestra la programación de un archivo que utilizamos como método de conexión para al servidor y consecuentemente a la base de datos.

Conexión.php

```

1  <?php
2  $host="localhost";
3  $user="root";
4  $password="";
5  $db="myapp";
6  $con = new mysqli($host,$user,$password,$db);
7
8  ?>

```

Este código muestra la programación de un archivo que utilizamos como método para el registro de los usuarios mediante un formulario de datos personales.

Registro.php

```
1 <?php
2
3 if(!empty($_POST)){
4     if(isset($_POST["username"]) &&isset($_POST["fullname"]) &&isset($_POST["email"]) &&isset($_POST["password"]) &&isset($_POST["password2"]){
5         if($_POST["username"]!="&& $_POST["fullname"]!="&&$_POST["email"]!="&&$_POST["password"]!="&&$_POST["password2"]!="&&$_POST["password2"]!= $_POST["password"]){
6             include "conexion.php";
7
8             $found=false;
9             $sql1= "select * from user where username=\"$_POST[username]\" or email=\"$_POST[email]\"";
10            $query = $con->query($sql1);
11            while ($r=$query->fetch_array()) {
12                $found=true;
13                break;
14            }
15            if($found){
16                print "<script>alert(\"Nombre de usuario o email ya estan registrados.\");window.location='../registro.php';</script>";
17            }
18            $sql = "insert into user(username,fullname,email,password,created_at) value ('$_POST[username]','$_POST[fullname]','$_POST[email]','$_POST[password]',now())";
19            $query = $con->query($sql);
20            if($query!=null){
21                print "<script>alert(\"Registro exitoso. Proceda a logearse\");window.location='../login.html';</script>";
22            }
23        }
24    }
25 }
26
27
28
29 ?>
```

Este código muestra la programación de un archivo que utilizamos como método para el acceso a los usuarios que se encuentran dentro de la base de datos mediante su correo y contraseña.

Login.php

```
1 <?php
2
3 if(!empty($_POST)){
4     if(isset($_POST["username"]) &&isset($_POST["password"])){
5         if($_POST["username"]!="&&$_POST["password"]!="&&$_POST["password"]!="&&$_POST["password"]!="&&$_POST["password"]!= $_POST["password"]){
6             include "conexion.php";
7
8             $user_id=null;
9             $sql1= "select * from user where (username=\"$_POST[username]\" or email=\"$_POST[username]\") and password=\"$_POST[password]\"";
10            $query = $con->query($sql1);
11            while ($r=$query->fetch_array()) {
12                $user_id=$r["id"];
13                break;
14            }
15            if($user_id==null){
16                print "<script>alert(\"Acceso invalido.\");window.location='../login1.html';</script>";
17            }else{
18                session_start();
19                $_SESSION["user_id"]=$user_id;
20                print "<script>window.location='../acceso.php';</script>";
21            }
22        }
23    }
24 }
25
26
27
28 ?>
```

Este código muestra la programación de un archivo que utilizamos como método para la desconexión de la cuenta del usuario mediante un botón cuando termino de usar el servicio.

Logout.php

```
1 <?php
2 session_start();
3 session_destroy();
4 print "<script>>window.location='../login.php';</script>";
5 ?>
```

CASOS DE PRUEBA

Proyecto:	Análisis para la implementación de un prototipo de rastreo de Transporta(URBAN Tampico)		
ID Caso de Prueba:	URTAM004		
Nombre Caso de Prueba:	Validación de seguridad en la contraseña		
Versión del Caso de Prueba:	1		
Fecha de Creación:	14/06/2016		
Fecha de Ejecución:	14/06/2016		
Autor del Caso de Prueba:	Abel Rivera Valdez		
Condición:	registro de usuario-ingreso de contraseña		
Flujo de pasos de la Prueba:	Num.Paso	Descripción del paso	Resultado Esperado
	1	ingresar a la pantalla de registro de usuario	mostrar la pantalla del registro de usuarios
	2	ingresar la contraseña	poder ingresar la contraseña y si no cumple los requisitos de validacion mostrar una advertencia
	3	pedir que la contraseña lleve mayusculas y numeros	que aparesca una ventana pidiendo que la contraseña lleve mayusculas y numeros
Decisión de Aprobación del Caso de Prueba:	seleccionar la opción resultante: Aprobó:___ Fallo: <u>X</u>		
Nombre y firma del Probador:	Abel Rivera Valdez		
Fecha de Aprobación del Caso de Prueba:	14/06/2016		

PROYECTO	Análisis para la implementación de un producto de rastreo de transporte (URBAN Tampico)			
ID CASO DE PRUEBA	URTAM003			
NOMBRE DE CASO DE PRUEBA	CP001			
VERSIÓN DEL CASO DE PRUEBA	V1			
AUTOR DEL CASO DE PRUEBA	HUGO ENRIQUE CECENES MORALES			
FECHA DE CREACIÓN	08/06/2016			
FECHA DE EJECUCIÓN	09/06/2016			
CONDICIÓN	REGISTRO E INICIO DE SESIÓN			
FLUJO DE PASOS DE LA PRUEBA	NO. PASO	DESCRIPCIÓN	RESULTADO ESPERADO	RESULTADO OBTENIDO
	1	Registrarse en la Aplicación	Datos registrados en la base de datos	No se insertan los datos
	2	Iniciar Sesión en la Aplicación	Acceder a la interfaz de rastreo	Como no se puede registrar, no se puede iniciar sesión
DECISIÓN DE APROBACIÓN DE CASO DE PRUEBA	APROBÓ FALLÓ X			
NOMBRE Y FIRMA DEL PROBADOR	HUGO ENRIQUE CECENES MORALES			

Proyecto:	Análisis para la Implementación de un prototipo de rastreo de Transporte(URBAN Tampico)			
ID Caso de Prueba:	URTAM002			
Nombre Caso de Prueba:	Verificación de registro en base de datos			
Versión del Caso de Prueba:	1			
Fecha de Creación:	14/06/2016			
Fecha de Ejecución:	14/06/2016			
Autor del Caso de Prueba:	Claudia Paola Castillo Abundis			
Condición:	Registro de usuario previo			
Flujo de pasos de la Prueba:	Num.Pas	Descripción del paso	Resultado Esperado	Resultado Obtenido
	1	Ejecutar el código de la página Web, el cual nos redirige al Login de los usuarios, donde presionamos el botón de "Registrar".	Acceder a la página de registro de los usuarios.	Accedí a la página Web del registro de los usuarios.
	2	Rellenamos los campos de datos para registrar al nuevo usuario a crear desde la interfaz Web y presionamos el botón de Registrar.	Si el registro se crea y conecta con la base de datos con éxito, nos redirige a la página de Login para Acceder con el usuario que creamos anteriormente.	Redirige a la página de Login.
	3	Entramos a la base de datos de LoginDB desde un nuevo Query y escribimos el comando para ver a todos los usuarios de la tabla Users(select * from Users).	El comando al ejecutarse muestra a los usuarios creados en la tabla con lo cual podemos verificar que el usuario que creamos mediante la interfaz Web se creó con éxito.	El usuario que creamos desde la interfaz se creó con éxito en la base de datos de nuestro servidor.
Decisión de Aprobación del Caso de Prueba:	seleccionar la opción resultante: Aprobó: <u>X</u> Fallo: <u> </u>			
Nombre y firma del Probador:	Claudia Paola Castillo Abundis			
Fecha de Aprobación del Caso de Prueba:	14/06/2016			

Proyecto:	Análisis para la Implementación de un prototipo de rastreo de Transporte(URBAN Tampico)			
ID Caso de Prueba:	URTAM001			
Nombre Caso de Prueba:	Verificación de Seguridad de Acceso a los Usuarios.			
Versión del Caso de Prueba:	1			
Fecha de Creación:	13/06/2016			
Fecha de Ejecución:	13/06/2016			
Autor del Caso de Prueba:	Alexander Gomez Enriquez			
Condición:	Login de Usuarios con los mismos datos de Acceso para la verificación de Seguridad.			
Flujo de pasos de la Prueba:	Num.Pas	Descripción del paso	Resultado Esperado	Resultado Obtenido
	1	Se registran 2 usuarios con los mismos datos de acceso pero con diferente correo electrónico (Por medio del correo se verifica la existencia del usuario en la base de datos) -INSERT INTO Users SELECT 'AlexGE', '12345', 'alexg@gmail.com', GETDATE(), NULL and SELECT 'AlexGE', '12345', 'alexg@hotmail.com', GETDATE(), NULL.	Users aparezcan los 2 usuarios con sus respectivos datos.	Aparecen los 2 usuarios nuevos que agregamos con sus respectivos datos.
	2	Ejecutamos la página de Login de acceso de usuarios.	Se espera que se abra el navegador Web al compilar el código de la página de Login.	La página web se ejecuta correctamente en el navegador.
	3	Llenamos los campos de acceso (Usuario: AlexGE, Contraseña: 12345) y Presionamos el botón de Acceder.	Se pretende Acceder a la página que es exclusiva del usuario, lo cual significa que accedí con éxito.	Accedimos a la página del Usuario con éxito.
	4	En la base de datos LoginDB en un Query visualizamos los usuarios con sus datos (Select * from Users) y nos enfocamos en el dato de la columna "LastLoginDate" y verificamos la última fecha de acceso de los usuarios que creamos anteriormente.	Se espera verificar cual de los 2 usuarios que creamos anteriormente se logró acceder.	Se accedí al último usuario que creamos con los mismos datos (alexg@hotmail.com), sin embargo el otro usuario (alexg@gmail.com) sigue existiendo en la base de datos, por lo cual el sistema funciona pero no de la manera
Decisión de Aprobación del Caso de Prueba:	seleccionar la opción resultante: Aprobó: <u> </u> Fallo: <u>X</u>			
Nombre y firma del Probador:	Alexander Gomez Enriquez			
Fecha de Aprobación del Caso de Prueba:	16/06/2016			

ARCHIVO ORIGINAL



Casos de prueba.xlsx

MANUALES DE USUARIO

Los siguientes archivos muestra la guía de uso detallada del uso del prototipo implementado en Web y en App Android.



Manual-de-Usuario
-Web.pdf



Manual-de-Usuario
-Urban-App.pdf

MANUAL TECNICOS

El siguiente archivo muestra de manera detallada los métodos a implementar para desarrollar y dar mantenimiento al prototipo Web, Android y la base de datos.



Manual Tecnico.pdf