

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



**Aplicación Móvil Para Consulta de Tiempos de Cruce en Puentes
Internacionales.**

Reporte Técnico de Investigación presentado por:

Juan Manuel Rodríguez Domínguez. Mat. 58922

David Simón de Santiago Camacho. Mat. 75692

Requisito para la obtención del título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

Profesor Responsable: Ing. Hugo Javier Brito Holguín

Mayo de 2013

Autorización de Impresión

Los abajo firmantes, miembros del comité evaluador autorizamos la impresión del proyecto de titulación

Elaborado por los alumnos:

Juan Manuel Rodríguez Domínguez. Mat. 58922

David Simón de Santiago Camacho. Mat. 75692

Ing. Hugo Javier Brito Holguín

Nombre

M.C. José Fernando Estrada Saldaña

Nombre

Declaración de Originalidad

Nosotros Juan Manuel Rodríguez Domínguez y David Simón de Santiago Camacho declaramos que el material contenido en esta publicación fue generado con la revisión de los documentos que se mencionan en la sección de Referencias y que el Programa de Cómputo (Software) desarrollado es original y no ha sido copiado de ninguna otra fuente, ni ha sido usado para obtener otro título o reconocimiento en otra Institución de Educación Superior.

Juan Manuel Rodríguez Domínguez

David Simón de Santiago Camacho

Dedicatoria

Dedico el presente documento a mis padres, a mi esposa por tener esta magnífica idea y ser un fuerte pilar en mi vida, a mis amigos por su apoyo constante como grandes seres humanos.

Juan Manuel Rodríguez Domínguez

Dedico este proyecto a mis padres, a mi hermana, cuñado, sobrino, quienes han influido mucho en el transcurso de mi carrera, a mis profesores y a mis amigos por entender cuando no puedo estar con ellos.

David de Santiago Camacho

Agradecimientos

Agradecemos a nuestros Profesores, que con su paciencia y conocimiento han fortalecido nuestra vida universitaria, agradecemos a nuestros padres, amigos, a quienes indirectamente han contribuido a este proyecto con ideas, consejos y opiniones, al Dr. Victor Morales por sus importantes contribuciones a este trabajo, y a Dios por permitirnos llegar con bien al término de esta obra y que los resultados de la misma le sean de utilidad a más personas.

Juan Manuel Rodríguez y David de Santiago Camacho

Índice

Autorización de Impresión	iii
Declaración de Originalidad	iv
Dedicatoria.....	v
Agradecimientos	vi
Índice	vi
Lista de Figuras	ixx
Lista de Tablas	x
Introducción	1
Capítulo 1. Planteamiento del Problema	2
1.1 Antecedentes.....	2
1.2 Definición del problema	7
1.3 Objetivos de la investigación	7
1.4 Preguntas de investigación.....	8
1.5 Justificación de la investigación.....	8
1.6 Limitaciones y delimitaciones de la investigación.....	9
Capítulo 2. Marco Teórico.....	11
2.1 Puentes internacionales entre México y Estados Unidos.....	11
2.2 Clima en Cd. Juárez – El Paso, Tx	12
2.3 Tipo de cambio dólar-peso mexicano.....	13
2.3.1 Casa de cambio	13
2.4 Ciudades fronterizas	14
2.5 Aplicaciones móviles.....	14
2.6 Plataformas móviles.....	17
2.6.1 Android.....	17
2.6.2 IOS	17
2.6.3 Blackberry OS	17
2.6.4 Symbian OS.....	18
2.7 Aplicaciones web.....	18
2.7.1 ASP.NET	18
2.8 Servicios web	19
2.9 XML.....	21
	vii

2.10 Bases de datos	22
2.10.1 Lenguaje SQL	22
2.10.2 Gestores de bases de datos	23
2.10.2.1 SQL Server	23
2.10.2.2 MySQL	23
2.11 Protocolos SOAP	24
2.12 Servidores	26
2.12.1 Servidores web	26
2.12.2 Servidor de bases de datos	26
Capítulo 3. Materiales y Métodos	27
3.1 Descripción del área de estudio	27
3.2 Materiales	27
3.3 Métodos	29
3.3.1 Web Service para Android	31
3.3.2 Servicio para pagina web	34
3.3.3 Captura de tiempos para estadísticas	35
3.3.4 Aplicación móvil	37
3.4 Definición de la métrica de evaluación de resultados de la aplicación	41
Capítulo 4. Resultados de la Investigación	42
4.1 Presentación de resultados	42
4.2 Análisis e interpretación de resultados	47
Capítulo 5. Discusiones, Conclusiones y Recomendaciones	50
5.1 Con respecto a las preguntas de investigación	50
5.2 Con respecto al objetivo de la investigación	51
5.3 Recomendaciones para futuras investigaciones	51
Referencias	53
Anexos	55
2.3 Web service para aplicación móvil	55
3.2 Web service para página web	81
3.3 Captura de tiempos para estadísticas	93
3.4 Código de aplicacion móvil	119
3.5 Base de datos	175

Lista de Figuras

Figura 1.1. Estadísticas de cruce. Best time to cross the border.....	4
Figura 1.2. Pantalla principal de Yahoo! tiempo	5
Figura 1.3. Pantalla principal de tipo de cambio en Tijuana	6
Figura 2.1. Promedio de temperaturas en Cd. Juárez de 1951 a 2011	13
Figura 2.2. SoundHound.....	16
Figura 2.3. Telemetro. Smart measure	16
Figura 2.4. La aplicación envía un mensaje de red al servidor que ofrece el servicio web	19
Figura 2.5. Resultado de la ejecución del servicio web solicitado por el cliente	20
Figura 2.6. Estructura de un mensaje SOAP	25
Figura 3.1. Móvil LG L7	27
Figura 3.2. Laptop Samsung R480.....	28
Figura 3.3. Laptop Gateway NV555	28
Figura 3.4. Mapa de desarrollo	30
Figura 3.5. Layout de la aplicación móvil	37
Figura 4.1. Pantalla principal de la aplicación.....	44
Figura 4.2. Pantalla que muestra la ubicación de las casas de cambio	44
Figura 4.3. Pantalla que muestra las estadísticas de los tiempos de cruce de cada puente	45
Figura 4.4. Pantalla que muestra la sección de alertas para los tiempos de los puentes	46
Figura 4.5. Pantalla que muestra el clima por hora de la ciudad	46

Lista de Tablas

Tabla 2.1. Ciudades fronterizas entre México y Estados Unidos	15
Tabla 4.1. Tabla de medición de tiempo de pantallas de aplicación móvil.....	42
Tabla 4.2. Tabla de diferencia entre diferentes fuentes.....	43
Tabla 4.3. Tabla de registros de tiempo para cruce San Ysidro	47

Introducción

El tiempo de espera en los cruces internacionales de las ciudades fronterizas es un factor que toma en cuenta una parte de la población usuaria de dichos cruces, este tiempo puede ser determinante de la decisión no solo del momento en el cual acude el usuario al puente internacional, sino también puede ser un factor para decidir acudir o no a él. Otro factor que en algunas personas puede influir a la hora de cruzar por un puente internacional es el clima, en Ciudad Juárez, una de las ciudades fronterizas objetivo de este proyecto, las temperaturas pueden ser muy extremas, sus cambios pueden ser muy marcados y hasta parecer impredecibles en algunos momentos, afortunadamente se cuenta con sistemas muy atinados y sofisticados de predicción del clima que pueden ayudar a tomar decisiones en las cuales el clima resulte ser un factor a tomar en cuenta.

Debido a que en Estados Unidos rara vez aceptan pesos mexicanos como moneda de intercambio, las personas que acuden desde México deben comprar la divisa americana para efectuar sus compras o pagos, es por eso que en las ciudades fronterizas como Ciudad Juárez existe una gran cantidad de casas de cambio para que la población en general compre o venda dólares americanos. El tipo de cambio de venta y de compra de dólares es un factor que se encuentra siempre en movimiento y las transacciones que se hacen en una casa de cambio están sujetas a este factor y es común que las personas que acuden a este negocio lo tengan en cuenta a la hora de efectuar sus compras o ventas de la divisa.

En el presente documento se trata el desarrollo de una herramienta para dispositivos móviles que proporcione información útil para los usuarios de puentes internacionales de los factores que se toman en cuenta al momento de hacer uso de ellos. Se obtuvo como resultado una aplicación de fácil uso que concentra en una sola herramienta tiempo de cruce, clima y tipos de cambio en las casas de cambio participantes en el proyecto.

Se agregaron algunas funcionalidades que podrían resultar útiles al usuario de la aplicación como estadísticas de tiempos de cruce, alertas para tiempos de cruce específicos, ubicación de casas de cambio en un mapa y, evolución por hora del clima de la ciudad.

Esta aplicación fue desarrollada para la plataforma móvil Android, y utiliza un servidor de base de datos de Microsoft SQL Server 2008.

Capítulo 1. Planteamiento del problema

1.1 Antecedentes

Las ciudades fronterizas que existen en México cuentan con 38 cruces fronterizos hacia Estados Unidos y son Tijuana, Mexicali, Nogales, Ciudad Juárez, Piedras Negras, Nuevo Laredo, Reynosa y Matamoros las ciudades más importantes entre ellas. En estas ciudades existe una gran cantidad de personas que hacen uso de estos cruces fronterizos y que deben acatar los tiempos de espera en las filas que existen para pasar hacia estados unidos. Después de los ataques terroristas del 11 de septiembre del 2001 en Estados Unidos las medidas de seguridad en los cruces internacionales se reforzaron, aumentando inevitablemente los tiempos de espera.

El volumen de vehículos que transitan los puentes internacionales en la frontera México-Estados Unidos es muy grande. Como ejemplo, de acuerdo a las estadísticas de la página de la Ciudad de El Paso Texas [1], en el cruce internacional de Santa Fe, en la frontera de Cd. Juárez con El Paso en el 2011 cruzaron 1,173,788 vehículos de pasajeros y en el 2010 fueron 1,239,464 vehículos por el mismo cruce.

Existen hoy en día un sistema de registro de tiempos de espera en los cruces fronterizos, proporcionado por el U.S. Customs and Border Protection de los Estados Unidos, a través de su página web [2]. En ella se concentra además del tiempo de espera en el puerto de entrada, las líneas de acceso abiertas, las líneas ready lane que son garitas exclusivas para personas que viajen con un documento de la Iniciativa de Viajes del Hemisferio Occidental (ó WHTI por sus siglas en inglés) compatible con Identificación por Radiofrecuencia y que agilizan el trámite de cruce. Además la página también registra tiempos de cruce peatonal y sus líneas abiertas.

Hay algunos usuarios de los cruces internacionales que toman en cuenta las predicciones climatológicas de la ciudad para planear su cruce. Un día de paseo, de compras o alguna actividad de esparcimiento podrían estar entre las razones para tomar en cuenta el estado del clima en el día de su viaje. Se debe recordar que una gran

cantidad de ciudades fronterizas se encuentran en zona de climas extremos, tan sólo en Ciudad Juárez, la temperatura mínima registrada es de -23°C y la máxima es de 47.3°C . En estas ciudades se presenta nieve, tormentas, granizo, lluvias torrenciales, vientos de gran velocidad, y altas temperaturas, que como se hizo mención existe un sector de la población que toma en cuenta estas circunstancias a la hora de hacer un viaje a través del cruce fronterizo de la ciudad.

Existen métodos para el cálculo del tiempo de los cuales se hacen uso para establecer los pronósticos climatológicos que se le ofrecen a la población, para hacer una predicción del tiempo los datos provenientes de todas las estaciones meteorológicas del país son vertidos en un mapa, llamado carta del tiempo, en promedio cada tres horas. El pronosticador traza las isobaras o curvas de presión en un gráfico y reconoce en ellas las distintas masas de aire, los sistemas de presión y los frentes. Esa carta podría verse como una fotografía del tiempo en todo el territorio que abarca el gráfico y se utiliza para hacer un diagnóstico de las condiciones meteorológicas. Paralelamente se anexan imágenes satelitales e imágenes de radar para dar una idea clara de la posición de los sistemas nubosos, frentes, áreas de buen y mal tiempo, etcétera. Una vez hecho un buen diagnóstico de la situación se comienza la tarea del pronóstico. Para ello se utilizan diversos modelos por computadora que toman la carta del tiempo actual y diagraman, teniendo en cuenta un gran número de fórmulas matemáticas. El pronosticador deberá interpretar en base a sus conocimientos esas cartas y elaborar el pronóstico. Está en la experiencia y habilidad del profesional la elaboración de un pronóstico certero.

La información de los pronósticos resultantes puede ser consultada en páginas web que el usuario considere de alta fiabilidad por lo atinado de sus predicciones. Las estaciones de radio y canales de televisión tienen sus fuentes de pronósticos climatológicos y son ellos los que a su vez dan a conocer estos resultados a los usuarios que sintonicen sus servicios.

Al momento se han descrito dos tipos diferentes de datos de los cuales los viajeros de los cruces internacionales hacen uso. Existe una gran cantidad de datos que éstos evalúan para tomar una decisión acerca del momento de cruzar a Estados Unidos, pero hay uno que sin duda alguna interesa a una gran cantidad de estos usuarios de cruces fronterizos, y es el tipo de cambio dólar americano/peso mexicano.

La paridad del dólar americano y el peso mexicano es establecida al igual que todos los tipos de cambio de monedas en el mundo. Se establece por el mercado

mundial de divisas en base a la demanda y oferta, es decir que por ejemplo un inversionista americano oferta de dólares cuando desea adquirir un bien en México y por consiguiente demanda pesos, el equilibrio en el mercado determina el valor de una moneda con respecto a la otra. La especulación, es decir la compra de cierta moneda por una certeza de que su valor subirá, también crea demanda y oferta por su contraparte de la moneda que tenga el comprador de la divisa. Pueden ser muchas las variables económicas que pueden llegar a afectar los precios de la moneda, por ejemplo la inflación que declare el banco central del país, y la depreciación o apreciación como medida de corrección automática de la balanza de pagos, que es la cuenta que registra el total de transacciones entre un país y el resto del mundo.

Existen muchas fuentes de información que nos indican la paridad del dólar con el peso, todos los bancos publican los valores de compra y venta en sus ventanillas, y las casas de cambio hacen lo propio en las suyas, para saber el precio de la divisa que se desea adquirir. Los usuarios de los cruces fronterizos en su mayoría deben adquirir dólares para gastar en Estados Unidos, los bancos y las casas de cambio son su opción para hacerlo.

Existen actualmente en el mercado de aplicaciones móviles algunas soluciones que son herramientas útiles a los usuarios de cruces internacionales. A continuación se describen algunas de las más importantes.

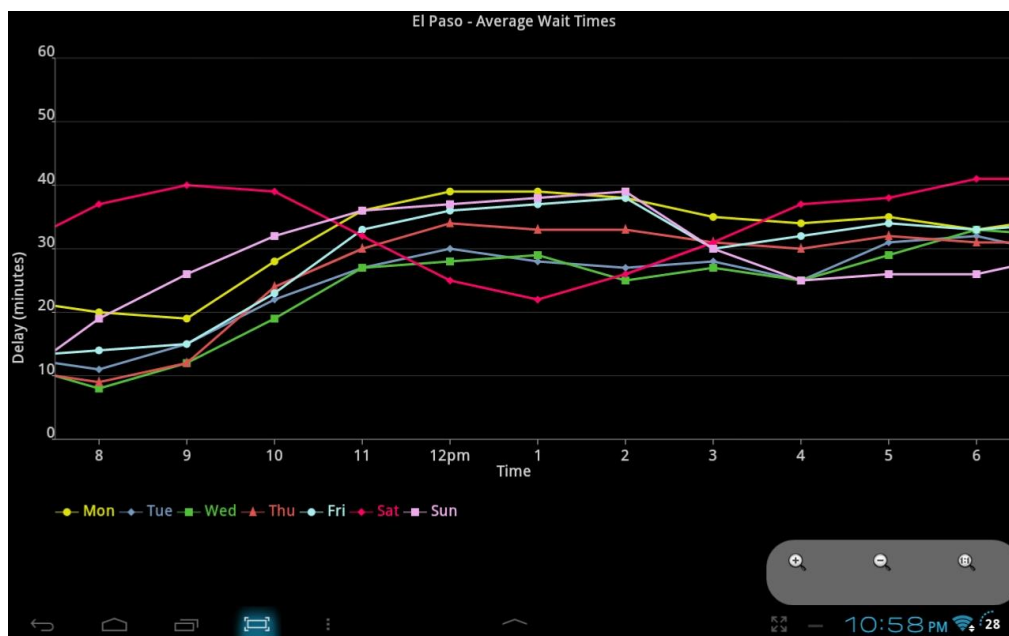


Figura 1.1 Estadísticas de cruce. Best time to cross the border

La figura 1.1 describe a una de las herramientas más completas que se ofrecen actualmente en Play Store (mercado de aplicaciones de Google Android) con relación a los cruces fronterizos [3]. Es propiedad de California Institute Of Telecommunications And Information Technology y tiene como finalidad proporcionar estadísticas del tiempo de espera en los puertos que se seleccionen; indica también los valores actuales de los tiempos de espera. Las configuraciones que ofrecen al usuario consisten en seleccionar los días de la semana que se desea que aparezcan en la tabla, y su visualización puede dar a simple vista los patrones que siguen los tiempos de espera con respecto al día y a la hora. Es una herramienta bastante útil si solo se desea analizar el dato del tiempo de espera en el cruce internacional. La figura 1.1 muestra estadísticas de tiempo de cruce tal como lo ofrece la herramienta



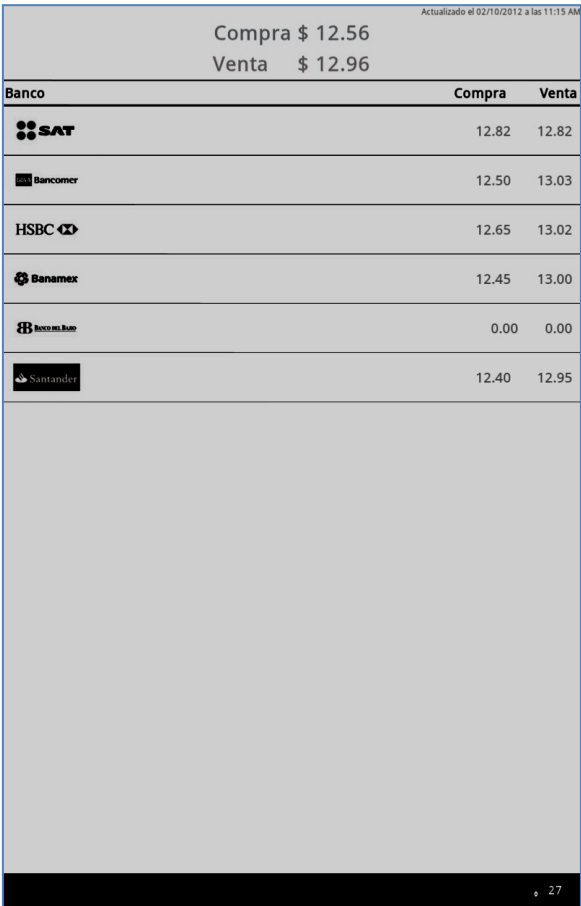
Figura.1.2. Pantalla principal de Yahoo! tiempo.

Yahoo! Tiempo, de Yahoo! Inc. Descrita en la figura 1.2 es una de las herramientas para visualización de pronósticos climatológicos más descargadas en el mercado de aplicaciones móviles [4], con más de 1,000,000 de descargas. Tiene una interface muy amigable, se encuentra en actualización constante de los datos y presenta en su pantalla principal las temperaturas máximas y mínimas esperadas para el día

actual, la temperatura en ese momento, la fase lunar que se puede observar en ese hemisferio y un pronóstico de los cuatro días siguientes, además de que ostenta una agradable imagen como fondo de la aplicación.

Entre las opciones de configuración, el usuario puede seleccionar entre °C y °F, además de la ciudad que se desea presentar como predeterminada al momento de ejecutar la aplicación.

Tipo de Cambio en Tijuana (figura 1.3), del desarrollador Iván Villareal [5], es la herramienta que se acerca más a la necesidad de conocer los tipos de cambio que se ofertan en los bancos en un momento dado. En su pantalla principal muestra los valores de compra y venta en pesos del dólar, no tiene configuraciones adicionales que pueda establecer el usuario y tampoco presenta otras fuentes de tipos de cambio que no sean de cinco bancos (Bancomer, HSBC, Banamex, Banco del Bajío y Santander) y del Servicio de Administración Tributaria de México.









Actualizado el 02/10/2012 a las 11:15 AM		
Compra \$ 12.56		
Venta \$ 12.96		
Banco	Compra	Venta
 SAT	12.82	12.82
 Bancomer	12.50	13.03
 HSBC	12.65	13.02
 Banamex	12.45	13.00
 Banco del Bajío	0.00	0.00
 Santander	12.40	12.95
27		

Figura. 1.3. Pantalla principal de tipo de cambio en Tijuana.

Encontraremos como conclusión del estudio de estas tres herramientas representativas de tiempos de cruce en puentes fronterizos, pronóstico climatológico y paridad de dólar/peso que ninguna ofrece la conjunción de estos tres tipos de datos. Teniendo como consecuencia que la persona que se vea en la necesidad de consultar más de uno de los tipos de datos mencionados debe hacerlo por medios diferentes o sin relación entre ellos, obligando al uso de varias aplicaciones, o de varias fuentes de información y de su intercambio como puede ser el escuchar una estación de radio y esperar al reporte del clima y consultar el estado del tiempo en el espacio del noticiero televisivo que se encuentre en ese momento, o de ejecutar la aplicación del pronóstico del clima y acudir al exterior de una casa de cambio para saber si la paridad dólar/peso se encuentra dentro del rango que la persona considere aceptable para adquirir la divisa. Sin embargo, a la fecha no existe una aplicación móvil que ofrezca al mismo tiempo esos tres datos para su visualización integral.

1.2 Definición del problema

Existe una porción de la población que hace uso de los cruces internacionales para los cuales sería más fácil tener la información de los tiempos de cruce, tipo de cambio actual, clima de la ciudad, consultar estadísticas del comportamiento en los tiempos de cruce así como los cambios de alza y baja del precio del dólar en una sola aplicación, en lugar de obtener esos datos de las fuentes que actualmente se encuentran distribuidas.

1.3 Objetivos de la investigación

Objetivo general:

Desarrollar un software que proporcione los tiempos de cruce en los puentes internacionales, la temperatura actual y el tipo de cambio con la finalidad de que los ciudadanos tengan una herramienta móvil que les proporcione dicha información de una manera eficiente.

Objetivos específicos:

- Presentar al usuario una herramienta con interfaz amigable.
- Proporcionar al usuario estadísticas de tiempos de cruce en la semana.
- Permitir al usuario programar alertas para rangos de tiempo de cruce específicos.
- Permitir al usuario comparar entre diferentes tipos de cambio ofertados en las casas de cambio.

Mostrar al usuario la ubicación en un mapa de la casa de cambio que elija.

1.4 Preguntas de investigación

- ¿Qué tan atractiva sería esta aplicación para usuarios de cruces fronterizos?
- ¿Qué fuentes de información existen para tiempos de cruce de puentes internacionales, tipo de cambio dólar-peso y clima de ciudades fronterizas?
- ¿Qué obstáculos se pueden presentar para la obtención y manejo de los datos de los cuales hace uso la aplicación?
- ¿Qué manejador de base de datos es el mejor para esta solución?
- ¿Qué plataforma de dispositivos móviles es la mejor para esta solución?
- ¿Qué Entorno de Desarrollo Integrado IDE es mejor para esta solución?

1.5 Justificación de la investigación

El desarrollo de una aplicación para un teléfono móvil que informe al momento los tiempos de cruce de los puentes internacionales hace que el usuario ya no tenga la necesidad de enviar un mensaje de texto con costo, visitar a través de un explorador de internet una página web o sintonizar una estación de radio y esperar a que el locutor mencione los tiempos de cruce, esto genera un ahorro de tiempo o de pasos para obtener dicha información; además del tiempo de cruce, el tipo de cambio es un asunto que generalmente interesa a los ciudadanos que cruzan hacia Estados Unidos, es por eso que otra de las funcionalidades de la aplicación a desarrollar incluye los precios de compra y venta de los principales bancos y casas de cambio de la ciudad. Por otro lado el pronóstico del clima local es un factor que un sector de la población toma en cuenta a la hora de planear una ida de compras o de diversión al país vecino. Es por eso que se hace

referencia a estos tres tipos de datos, el tiempo de cruce, el tipo de cambio y el clima en esta aplicación.

1.6 Limitaciones y delimitaciones de la investigación

1.6.1 Limitaciones

Una de las características más importantes de la aplicación móvil es el tiempo de cruce en los puentes internacionales, esta información sólo tiene una fuente oficial, que es la página de U.S. Customs and Border Protection, esto constituye la primera limitación pero no así la más importante. Debido a que la información fluye desde una sola fuente podría eso considerarse como un riesgo debido a que, en caso de que dicha página deje de funcionar, la información del cruce de los puentes internacionales no estaría disponible para la aplicación móvil ni para ninguna otra solución que haga uso de esa fuente de información; sin embargo debido a que dicha página web contiene la información para todos los puertos de entrada de Estados Unidos cuenta con un alto grado de fiabilidad y con un porcentaje de tiempo no disponible muy bajo.

Otra limitación que se presenta para el desarrollo de dicho proyecto es el poco conocimiento del área de desarrollo en aplicaciones móviles con el que contamos (los desarrolladores del proyecto), y que sólo nos ubica en una limitación temporal, debido a que, eventualmente, gracias a la información que hoy en día existe podremos (los desarrolladores del proyecto), llegar a acumular satisfactoriamente los conocimientos necesarios para culminar esta investigación.

1.6.2 Delimitaciones

La aplicación muestra el último tiempo de espera insertado en la base de datos, esto delimita la actualización del tiempo de espera a la última actualización de la página oficial de tiempos de cruce y a la última inserción en la base de datos que utiliza la aplicación.

El aviso que se ofrece para hacer del conocimiento del usuario que el tiempo de espera en algún cruce seleccionado está por debajo del configurado en la aplicación, está delimitado a una alerta propia del dispositivo móvil, es decir lo equiparable a un

sonido de alarma, no a un mensaje de texto, que también podría ser entendido como una alerta.

El historial de tiempos de cruce tiene un periodo de muestreo de una semana, es decir que sólo la semana anterior puede ser mostrada en dicho historial. Esta funcionalidad no está pensada para mostrar tendencias anuales ni mensuales. Por lo tanto en la base de datos solo se almacenará el periodo de una semana en registros de tiempo.

La actualización de los tiempos de cruce se hará en intervalos de 10 minutos. Esto significa que aunque el tiempo de actualización de la página oficial de tiempos de cruce no se ofrece, lo más retrasado que puede llegar a estar la información contenida en la base de datos es de 10 minutos con respecto a la de la página oficial.

Un aspecto importante es el hecho de que los usuarios de cruces fronterizos requieren de un dispositivo móvil con una plataforma apropiada para el funcionamiento de la aplicación, es decir Android. Es necesario también mencionar el hecho de que la aplicación funciona bajo el esquema cliente-servidor. Es decir, que necesita una conexión al servidor que solo puede establecerse contando con conexión de datos en el dispositivo móvil así como de una conexión a internet del servidor que cuente con la información.

Capítulo 2. Marco Teórico

En este capítulo se hace una breve referencia a los temas que sirven de base para la construcción del proyecto en sus dos divisiones, los elementos web y los elementos de aplicación móvil.

Los temas del dominio general como el clima, frontera y algunos puntos financieros como la paridad dólar – peso mexicano son importantes en el ámbito del presente proyecto, debido a que el planteamiento del problema se encuentra contextualizado en estos elementos económicos y sociales. La solución a ese problema se encuentra precisamente en la tecnología.

Base de datos es un concepto puntualizado en varias partes del proyecto, debido a que uno de los elementos que se encuentran alojados en el servidor es precisamente una base de datos.

También se hace una pequeña referencia a servidores, ya que tanto los componentes web así como la aplicación móvil obtienen e ingresan datos a un servidor.

2.1 Puentes internacionales entre México y Estados Unidos

México al ser un país fronterizo con Estados Unidos, cuenta con varios puertos distribuidos en los estados que conforman la franja fronteriza, los cuales permiten a los ciudadanos tener el acceso con el vecino país para realizar sus viajes por motivos de compras, visitas familiares, turismo, entre otros.

Existe una gran cantidad de puentes de cruce internacionales entre México y Estados Unidos. Estos son usados por los habitantes de las ciudades fronterizas que cuentan con una identificación expedida por el Consulado Americano, la cual los hace personas aptas para tener el acceso a su país.

Los estados de la parte de México que son frontera con el vecino país son: Baja California, Sonora, Chihuahua, Coahuila, Nuevo León y Tamaulipas. Por su parte los estados americanos con frontera en México son: California, Arizona, Nuevo México y Texas. [6]

La frontera considerada dentro de las más importantes de México desde 1848 es la de Ciudad Juárez y El Paso, Texas.

Esta ciudad cuenta con 4 puentes internacionales para acceder a los Estados Unidos:

- Puente Internacional Paso del Norte: conocido también con el puente Santa Fe, ubicado en la calle Juárez en el centro de la ciudad. Este puente cuenta con un solo sentido México-Estados Unidos.
- Puente Internacional Reforma: ubicado en la calle Lerdo en el centro de la ciudad con sentido de Estados Unidos-México.
- Puente Internacional Córdova de las Américas: Ubicado en la calle de las Américas en el Chamizal. Cuenta con los dos sentidos.
- Puente Internacional Zaragoza-Ysleta: Ubicado en la calle Waterfill de la ciudad y este cruce también cuenta con los dos sentidos de tráfico.[3]

2.2 Clima en Cd. Juárez - El Paso, Texas.

Ciudad Juárez tiene un clima extremo, debido a que su superficie es desértica. Y la diferencia entre sus temperaturas en función a las estaciones del año es muy grande; la media del mes de enero es de 5.3 °C y la media de julio es de 27.3 °C es decir una variación térmica de 22 °C entre la media de los meses más cálidos y fríos.

En el verano la temperatura varía de los 35 °C a los 18 °C siendo más caluroso el día y más fría la madrugada.

Se registran lluvias en promedio de 40 a 50 mm mensuales durante los meses más lluviosos, que son julio y septiembre.

En el invierno las temperaturas oscilan de los 14 °C a los -1 °C y las principales nevadas se presentan entre los meses de diciembre y febrero.

Recientemente en 2011 un frente frío muy fuerte hizo disminuir la temperatura a -18.5 °C rompiendo el récord previo para el mes de febrero de -17 °C registrado el 13 de febrero de 1963 en la ciudad. En la figura 2.1 se muestran los promedios de temperaturas registradas en Cd. Juárez de 1951 a 2011 por meses [7].

Parámetros climáticos promedio de Ciudad Juárez (CILA - 1.135 msnm)													
<ul style="list-style-type: none"> • Normales : 1951-2010 • Extremas : 1951-2011 													
Mes	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Anual
Temperatura máxima registrada (°C)	26.0	29.5	33.0	38.0	39.0	44.0	47.3	39.0	39.0	35.0	32.0	24.5	47.3
Temperatura diaria máxima (°C)	13.0	15.9	20.3	25.8	30.5	34.6	35.0	33.9	30.2	25.4	18.8	13.9	24.8
Temperatura diaria promedio (°C)	5.3	8.2	11.8	17.2	21.5	25.4	27.3	26.2	22.8	17.1	10.7	6.6	16.7
Temperatura diaria mínima (°C)	-2.3	0.5	3.4	8.5	12.5	16.2	19.6	18.5	15.4	8.9	2.5	-0.8	8.6
Temperatura mínima registrada (°C)	-23.0	-18.5	-13.0	-5.0	1.0	5.0	10.0	10.0	7.0	0.0	-9.0	-12.0	-23.0
Precipitación total (mm)	9.0	10.1	5.0	4.5	5.4	19.7	40.0	48.4	37.6	17.1	11.7	20.2	228.7
Nevadas (cm)	7.4	2.3	1.3	1.0	0	0	0	0	0	0.5	0.5	7.9	20.8
Días de lluvias (≥ 1 mm)	2.9	2.6	2.0	0.9	1.1	3.0	5.8	6.2	4.5	2.8	2.8	3.5	38.1
Días de nevadas (≥ 0.25 cm)	1.3	0.6	0.3	0.1	0	0	0	0	0	0.2	0.2	1.3	4.0

Figura. 2.1. Promedio de temperaturas En Cd. Juárez de 1951 a 2011.

2.3 Tipo de cambio dólar-peso mexicano

Como en cualquier ciudad del mundo, tanto como México y Estados Unidos, cuentan con su propia moneda para manejar la economía de cada uno. Una frente a la otra tiene un valor distinto dependiendo en donde sea utilizada. La moneda mexicana no se puede utilizar en los Estados Unidos, por otro lado la moneda americana si es aceptada en México con un valor arriba de lo que vale un peso mexicano. En las últimas dos décadas, el valor de dólar en México ha cambiado constantemente llegando desde los 3 pesos hasta su mayor valor de 15 pesos.

2.3.1 Casas de cambio

Una casa de cambio es un establecimiento en el cual un cliente llega a realizar el cambio de moneda que sea de su interés en el momento de su transacción. La persona interesada puede llegar al establecimiento con el interés de comprar dólares ya que realizará un viaje al vecino país y necesita llevar la moneda que en este se utiliza o de lo contrario puede llegar con la moneda americana y cambiarla por su equivalente en moneda mexicana para así realizar sus transacciones en nuestro país.

Dentro de estos negocios también se pueden adquirir seguros para automóviles, los cuales son necesarios para viajar a Estados Unidos ya que es un requisito que el país

pide para poder circular por las calles de la ciudad, ya que de no tener un seguro de manejo esto le traerá al ciudadano una multa en el país americano.

Por lo general, las casas de cambio se encuentran ubicadas a los alrededores de los puentes internacionales, para que sea mucho más fácil el acceso para los viajeros y realizar su cambio de moneda más rápido y sin tener que buscar por lados lejanos y tener una pérdida de tiempo. También se pueden encontrar en centros comerciales, ya que muchas veces se realizan compras dentro de estos y se tiene que hacer un cambio de moneda.

2.4 Ciudades fronterizas

La frontera entre México y Estados Unidos es la frontera que tiene mayor número de cruces legales en el mundo, en 1994 era de 34 millones y en 2002 llegó a los 50 millones de cruces peatonales al año. En la tabla 2.1 se mencionan las principales ciudades fronterizas entre México y Estados Unidos, ellas totalizan un número de 68 puertos fronterizos.

2.5 Aplicaciones móviles

Con el paso del tiempo las tecnologías móviles han ido avanzando, desde un simple celular, solo para hacer llamadas con una pantalla monocromática hasta llegar a lo que hoy es conocido como los Smartphone o teléfono inteligente.

Una aplicación móvil es un tipo de aplicación diseñada para ser ejecutada por el sistema operativo de un teléfono celular o una tableta, proporcionan en la mayoría de los casos servicios parecidos a los brindados en una computadora tradicional, son en lo general ligeras y de fácil uso, aplicaciones de mensajería instantánea, música, videos, noticias y juegos son muy comúnmente encontradas instaladas en los teléfonos inteligentes.

Ciudades Fronterizas (EU)	Ciudades Fronterizas (México)
San Diego, California.	Tijuana, Baja California.
Tecate, California.	Tecate, Baja California.
Calexico, California.	Mexicali, Baja California.
Andrade, California.	Los Algodones, Baja California.
San Luis, Arizona.	San Luis Rio Colorado, Sonora.
Lukeville, Arizona.	Sonorita, Sonora.
Sassabe, Arizona.	Sasabe, Sonora.
Nogales, Arizona.	Nogales, Sonora.
Naco, Arizona.	Naco, Sonora.
Douglas, Arizona.	Agua Prieta, Sonora.
Antelope Wells, Nuevo México.	El Berrendo, Chihuahua.
Columbus, Nuevo México.	Puerto Palomas, Chihuahua.
Santa Teresa, Nuevo México.	San Gerónimo, Chihuahua.
El Paso, Texas.	Ciudad Juárez, Chihuahua.
Presidio, Texas.	Ojinaga, Chihuahua.
Del Rio, Texas.	Ciudad Acuña, Coahuila.
Eagle Pass, Texas.	Piedras Negras, Coahuila.
Webb Co., Texas.	Colombia, Nuevo León.
Laredo, Texas.	Nuevo Laredo, Tamaulipas.
Misión, Texas.	Reynosa, Tamaulipas.
Hidalgo, Texas.	Reynosa, Tamaulipas.
Pharr, Texas.	Reynosa, Tamaulipas.
Donna, Texas.	Rio Bravo, Tamaulipas.
Progreso Lakes, Texas.	Nuevo Progreso, Tamaulipas.
Brownsville, Texas.	Heroica Matamoros, Tamaulipas.

Tabla 2.1 Ciudades fronterizas entre México y Estados Unidos

Para el desarrollo de una aplicación móvil, se deben tomar ciertos aspectos para que al finalizar esta, se obtenga un resultado satisfactorio y la funcionalidad sea 100% correcta. Algunos de estos aspectos a tener en cuenta deben ser: la plataforma en la cual se va a correr la aplicación y para que móviles está disponible esta plataforma (IOS, Android, Blackberry o Symbian). El lenguaje que esta plataforma soporta y en el cual se

va a desarrollar la aplicación. Las herramientas que nos facilitaran el desarrollo de esta y que están dentro de los programas que vamos a utilizar para realizarla. [4]

Algunas aplicaciones móviles desarrolladas para la plataforma Android de gran aceptación son:

SoundHound (figura 2.2). Que permite utilizar el micrófono del móvil para analizar una canción, compararla contra una base de datos y devolver al usuario información de la canción, como su título, cantante, y otros datos importantes, además de ligas a los videos o letras de la canción.

Telémetro Smart Measure (figura 2.3). Permite al usuario a través de trigonometría calcular la altura aproximada de objetos que capture mediante la cámara del móvil, así como la distancia a la que se encuentra el observador del objeto.

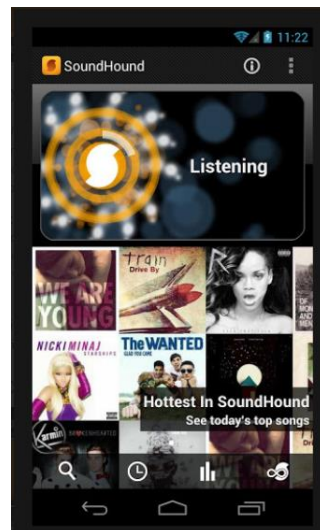


Figura. 2.2. SoundHound.



Figura 2.3. Telémetro. Smart measure.

2.6 Plataformas móviles

Con el paso del tiempo los dispositivos móviles han ido adquiriendo funcionalidades específicas dependiendo la marca del equipo, las funcionalidades y el desempeño del mismo, para que dependiendo de las utilidades de este manejar una plataforma que se adecue a las necesidades para su desarrollo y mejor uso.

Una plataforma viene siendo, dentro del dispositivo, el sistema operativo que hace funcionar al móvil y ofrecer la mejor tecnología al usuario para que satisfaga las necesidades por las cuales adquiere este producto.

2.6.1 Android

Android es una plataforma para dispositivos móviles la cual soporta aplicaciones realizadas en Java con un núcleo en Linux. Maneja software libre para dar a los usuarios la oportunidad de crear sus aplicaciones y ponerlas en el mercado. Su manejador principal es Google, el cual creó esta plataforma para los dispositivos móviles [8].

Dentro de los dispositivos que usan esta plataforma se encuentran: Motorola, Sony Ericsson Xperia, LG, HTC, entre otros, los cuales ofrecen este sistema operativo al usuario para obtener aplicaciones y una experiencia mucho más avanzada en tecnología.

2.6.2 IOS

IOS es la plataforma que ofrece la compañía Mac Apple, para sus dispositivos ya sean reproductores de música Ipod o celulares Iphone. Cuenta con miles de aplicaciones para los celulares ipods, pero no permite la instalación de aplicaciones no oficiales. Para poder realizar esta acción existe un programa llamado Jailbreak el cual permite hacerlo libre y así poder tener acceso a aplicaciones creadas por otros usuarios.

2.6.3 Blackberry OS

Es el sistema operativo desarrollado para la gama de celulares de esta compañía. Es un sistema desarrollado para uso profesional para gestor de contactos y correo electrónico. Soporta lenguaje Java MIDP.

2.6.4 Symbian OS

Sistema operativo estándar y abierto para dispositivos móviles. Algunos de los usuarios son: Motorola, Nokia, Samsung y Sony Ericsson. Entre sus características se encuentra: una eficiente memoria RAM y ROM, está basado en micro Kernel, las aplicaciones usan sus propios procesos y su memoria designada, manejo fiable de datos y permite la conectividad entre dispositivos por Bluetooth.

2.7 Aplicaciones web

Una aplicación web es una forma de aplicación del tipo cliente – servidor, el cliente es el navegador de internet que utiliza el usuario y el servidor es el lugar en el que se encuentra almacenada la página que está visualizando el navegador, es decir el servidor web.

El cliente web está formado por código HTML y código en lenguajes de tipo script que puede ser ejecutado en el explorador como VBScript, JavaScript, etcétera.

El servidor, que en este caso es el servidor web, atiende las peticiones realizadas por el cliente a través del protocolo HTTP.

2.7.1 ASP.NET

Una aplicación web que invoca a una función almacenada en el servidor en realidad está llamando a una función contenida en un DLL (Dynamic Link Library) y en su totalidad el sistema Active Server Pages (ASP) es una DLL del Internet Server Application Program Interface (ISAPI), que es una serie de funcionalidades que permite responder de manera dinámica a peticiones HTTP en vez de sólo regresar archivos HTML estáticos.

Esta DLL ASP interpreta previamente el código de un archivo .asp como un script que debe ser ejecutado en el servidor, para eso llama a otro DLL que en base al

lenguaje en el que fué escrito dicho código lo ejecuta y su resultado es manejado por el servidor, para ser enviado al cliente que originó la ejecución.

ASP.NET es un Framework de Microsoft que permite a los programadores la construcción de Aplicaciones Web en lenguajes soportados por la plataforma .NET que son C#, Visual Basic .NET, Delphi, C++, F#, J#, Perl, Python, Fortran, P# , Cobol y PowerBuilder.

Los archivos de ASP.NET son llamados formularios web y en ellos se encuentran los controles que son ejecutados en el servidor y los controles que el programador asigna con código estático y dinámico. Estos archivos llevan por extensión ASPX.

Cabe mencionar que las aplicaciones desarrolladas en ASP.NET solo funcionan en servidores de Microsoft IIS (Internet Information Services) [9].

2.8 Servicios web

Debemos imaginar a un servicio web como una función, esta puede o no recibir parámetros iniciales, tiene como finalidad el ejecutar una instrucción y eventualmente devolver un resultado; idealmente lo mismo hace un servicio web, ejecuta una instrucción en base a un parámetro de entrada, al termino de esa ejecución devuelve un resultado, la diferencia estriba en que la función que utiliza una aplicación reside localmente, en la máquina que ejecuta dicha aplicación y el servicio web reside en un servidor remoto.

En el momento en el que la aplicación llama al servicio web como se muestra en la figura 2.5, se envía un mensaje a través de la red (esto supone que para que un servicio web sea útil, debe residir en un servidor con acceso a la red) a la dirección del servidor destino.

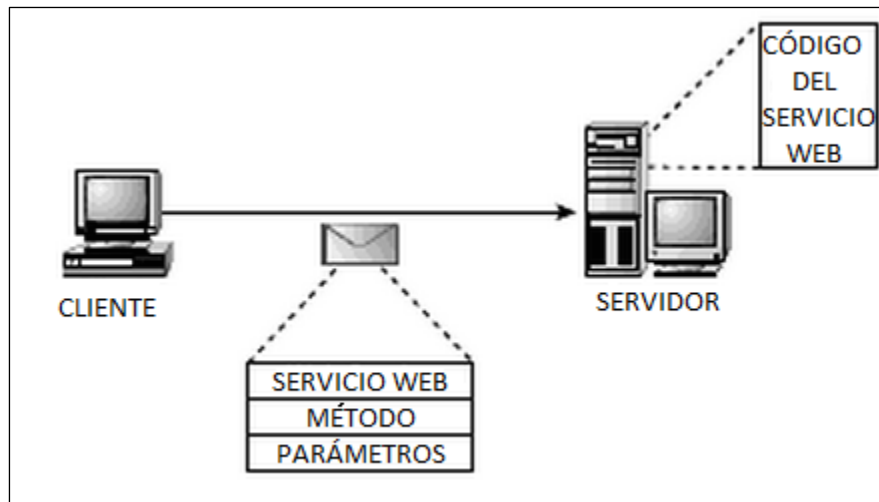


Figura 2.4. La aplicación envía un mensaje de red al servidor que ofrece el servicio web.

Después de que el servidor ejecuta la instrucción que contiene el código del servicio web, responde con un mensaje a través de la red a la aplicación cliente. Como se muestra en la figura 2.6.

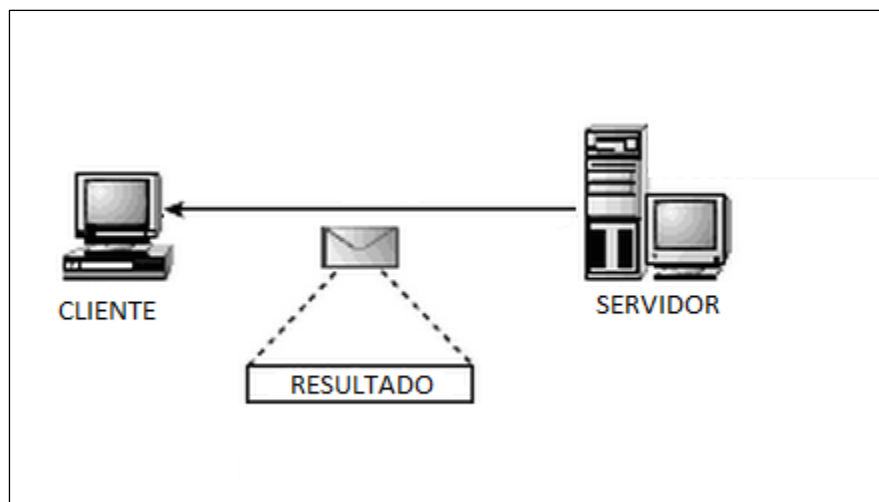


Figura 2.5. Resultado de la ejecución del servicio web solicitado por el cliente.

Debido a que los servicios web son llamados y responden a través de mensajes por la red, la carga que tenga ésta en el momento de la llamada al servicio o la carga de la red al momento de enviar la respuesta, además de la carga de operaciones del servidor en el momento de la ejecución hace que la respuesta de un servicio web sea mucho más lenta que la respuesta de una función que resida en la máquina que ejecute la aplicación.

Es por esto que el uso ideal de un servicio web es aquel que requiere de datos almacenados en lugares remotos, o de procesos que no se desean detallar en la

aplicación que ejecuta el usuario, por ejemplo una autenticación de tarjeta de crédito, existencias en inventarios, etcétera [9].

2.9 XML

XML significa por sus siglas Lenguaje Extensible de Marcas, el lenguaje de marcas más conocido es el HTML o Lenguaje de Marcas de Hipertexto, que es un lenguaje que tiene como finalidad dar un formato visual a la información; es decir que a través de un analizador o programa se le dará un formato a el texto escrito en el lenguaje HTML para crear una página web. XML es usado para crear una estructura de la información que se desea transmitir de ahí que, varias páginas web puedan tener la misma estructura pero no el mismo formato visual. Entonces, HTML es usado para desarrollar formatos, y XML para desarrollar estructuras.

La creación de una marca en el lenguaje XML se deja a consideración del programador, es decir, que es él quien decide que marcas crear XML según las necesidades que él decida cubrir con la implementación de su desarrollo.

Para que los programas analizadores de XML conozcan si la información contenida en el XML está estructurada de manera correcta se encuentran los archivos DTD, o Definición de Tipo de Documento, que contienen la información detallada de que marcas se van a usar en el documento XML, entonces así se pueden validar varios documentos XML y a través de un programa analizador se puede extraer la información de los XML con la confianza de que la estructura de los XML es la misma, y que se está extrayendo el mismo tipo de información de cada uno de los elementos del XML.

La ventaja de utilizar XML es que no es relevante el tipo de aplicación que lea o analice un XML, siempre y cuando sepa qué es lo que está buscando dentro del documento XML ya sea una aplicación de Windows, Linux, etc.

Ya que está escrito en texto plano puede ser muy fácil enviar una gran cantidad de datos de una base de datos a otra, además la plataforma en la que se ejecute el analizador de archivos XML no afecta a la estructura del archivo. Y es posible convertir casi cualquier tipo de información de tablas de bases de datos, de hojas de cálculos, información de clientes, artículos, calificaciones etcétera, en un documento XML [10].

2.10 Bases de datos

Una base de datos es una entidad en donde se pueden almacenar datos de forma estructurada, es decir un archivo o varios que contienen información ordenada en tablas, estas tablas almacenan registros compuestos de campos. Los campos de una tabla definen los tipos de datos que contienen, es decir un campo de tipo imagen, solo contiene ese tipo de dato almacenado. En una tabla todos los registros o datos almacenados se encuentran en los campos definidos al momento de diseñar la tabla.

Los sistemas de archivos guardan varias copias de los datos en archivos diferentes, esto significa que se utilice más espacio del que se debería usar, a diferencia de esto, una base de datos almacena los datos en un solo archivo o varios según sea la necesidad o diseño de la base, pero aun así se reduce la redundancia de los datos, es decir, se hacen solo las copias de los datos que debe haber redundantes. Gracias a esto, existe una menor inconsistencia de los datos. Esto significa que si existe un dato que debe ser actualizado y este no tiene más copias dentro de la base, al momento de ser actualizado se garantiza que todos los accesos que se hacen al dato muestren el mismo valor. A diferencia de un sistema de archivos en donde se actualice el dato de un archivo, pero sus copias en otros archivo no [11].

2.10.1 Lenguaje SQL

Para poder realizar consultas a una base de datos existe un lenguaje utilizado muy comúnmente, el lenguaje SQL.

En 1970 E.F. Codd publica su libro “Un modelo de Datos Relacional para Grandes Bases de Datos Relacionales” marcando las pautas a seguir de las bases de datos relacionales y dos años después, en los laboratorios de IBM se crea el Standard English Query Language al que se le llamó SEQUEL. Posteriormente se convierte en un estándar de las organizaciones ISO y ANSI con el nombre de SQL.

El Lenguaje SQL define, manipula y controla una base de datos relacional. A diferencia de los lenguajes procedimentales, en donde hay que especificar cómo hay que realizar la operación, SQL es un lenguaje declarativo, es decir solo hay que declarar la acción a realizar en la base de datos.

Se compone de comandos, cláusulas, funciones y operadores que se combinan en las consultas para hacer las modificaciones o creaciones de las bases de datos.

Básicamente este lenguaje es aceptado en los diferentes productos de base de datos, sin embargo cada fabricante o diseñador de esos productos extiende el lenguaje para la conveniencia de su software.

SQL se compone de dos tipos de comandos: DDL (Data Definition Language), que crea y define nuevas bases de datos, índices y campos; y DML (Data Manipulation Language) que permite el filtrado, orden, inserción y extracción de datos de las consultas que se generen [12].

2.10.2 Gestores de bases de datos

Un gestor de bases de datos es un sistema que le permite a un usuario la creación, modificación y utilización de una base de datos. En general es la interface que le permite interactuar con las bases de datos. Algunos gestores más conocidos son: Oracle, MySQL, MS SQL Server, entre otros.

Los gestores de bases de datos presentan las siguientes características:

Redundancia: es la repetición de un dato que aparece en varios lugares o cuando un dato no se modifica en todos los lugares donde debe de aparecer.

Rigidez de búsqueda: se maneja conforme al uso que se le dé al fichero, si por algún motivo se desea ingresar de otro modo esto resultara mucho más lento.

Dependencia de los programas: la información que esta base de datos contiene depende del programa en que se realice, en el caso de que se quiera usar en otro programa se tendrá que modificar.

Problemas de seguridad y confidencialidad: en confidencialidad se debe de tener en cuenta quien puede consultar ciertos datos, mientras que en seguridad se cuida que no cualquiera pueda modificar los datos introducidos [13].

2.10.2.1 SQL Server

Es un sistema de gestión de bases de datos relacionales que soporta el trabajo con grandes cantidades de datos especializado en aplicaciones comerciales y sitios web. Está diseñado para trabajar con dos tipos de bases de datos:

- OLTP (Online Transaction Processing): bases de datos con gran cantidad de usuarios conectados.
- OLAP (Online Analytical Processing): bases de datos con grandes cantidades de datos que sirven para tomar decisiones [14].

2.10.2.2 MySQL

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, es un sistema de software libre licenciado por la GNU GPL [15].

Se utiliza en aplicaciones web, en plataformas y herramientas de seguimiento. Es una base de datos de rápida lectura usando el motor MyISAM.

- Dentro de sus características están:
- Amplio subconjunto del lenguaje SQL.
- Disponibilidad de plataformas y sistemas.
- Posibilidad de selección de mecanismo de almacenamiento.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

2.11 Protocolo SOAP

La creciente necesidad de conectar aplicaciones alojadas en diferentes máquinas empezó a dejar al descubierto la falta de interoperabilidad entre los diferentes protocolos de encapsulación de código como CORBA (Common Object Request Broker Architecture) o COM (Component Object Model) populares desde el año 1990 para invocar a métodos en diferentes lenguajes de programación. Esto hizo que empezaran a aparecer protocolos de comunicación como DCOM (Distributed COM) para llamar remotamente a un objeto COM o IIOP (Internet Inter-ORB Protocol) para llamar remotamente a un procedimiento CORBA a través de la red. Sin embargo estos protocolos no son compatibles entre sí.

El Protocolo SOAP (Simple Object Access Protocol) se dio a conocer en 1999 como un desarrollo en conjunto de Microsoft Corp., DevelopMentor Inc. y

Userland Software Inc. Como un protocolo que intenta resolver los problemas de interoperabilidad especificando la forma del mensaje que invoca a los métodos remotos.

Funciona para la transmisión de texto en base a HTTP y en XML para la codificación de los datos. Esto asegura que cualquier cliente con un navegador pueda realizar una petición de conexión a un servidor y que los datos enviados en el formato XML puedan ser leídos ya que XML es un estándar aceptado para intercambio de datos actualmente.



Figura 2.6. Estructura de un mensaje SOAP.

Como se muestra en la figura 2.7 un mensaje SOAP consta de 3 partes: envelope (envoltura), header (Cabecera) y body (Cuerpo).

La envoltura es el elemento que define al documento XML como un mensaje SOAP.

La cabecera contiene información específica de la aplicación, como autenticación del mensaje SOAP y es un elemento opcional.

El cuerpo contiene el mensaje, en donde se hacen las llamadas y como se deben manejar las respuestas de los procedimientos remotos [16] [17].

2.12 Servidores

Un servidor es un computador que ofrece la facilidad de conexión de otras computadoras en la red para adquirir un servicio, el cual está alojado en estos esperando a la petición del cliente para ser mostrado [18].

2.12.1 Servidores web.

Un servidor web es el lugar en donde se almacenan todas las páginas existentes que circulan por la red, a las cuales accedemos diariamente cuando navegamos por internet. Su función es la de esperar una petición del usuario a la cual el servidor responde con una página web o algo relacionado con los comandos introducidos.

Existen dos tipos de servidores web: el dedicado, el cual solo se encarga de dar alojamiento a un solo cliente y el compartido, que se puede utilizar para varios clientes y almacenar diferente información.

Existen varios servidores web de los que destacan: Apache, Microsoft IIS, Sun Java System Web Server, entre otros. [19]

2.12.2 Servidor de bases de datos

Los servidores de bases de datos surgen por la necesidad de las empresas a manejar una gran cantidad de datos al mismo tiempo de que se desea compartirlos con varios usuarios.

Un servidor de bases de datos es un sistema bajo la arquitectura cliente/servidor que ofrece servicios de gestión, administración y protección de los datos. Su arquitectura está basada por tablas las cuales se conforman por campos en los cuales se introduce la información que describen su contenido y se comunican entre sí por medio de claves. Todo esto se puede lograr gracias al lenguaje más utilizado de la actualidad llamado SQL.

Como es sabido que en las bases de datos se maneja información muy importante de las empresas, estos servidores ofrecen 4 tipos de seguridad para resguardarla de cualquier intruso que desee tener un acceso no lícito de la información.

La seguridad de acceso: se implementa de dos maneras, a nivel de sistema operativo donde comprueba la validez del acceso a los datos almacenados o la de modo mixto, en la cual el servidor es el que se encarga de la validez del acceso.

La seguridad a nivel de objetos: es la que se encarga de validar el acceso a creación y modificación de datos.

La seguridad a nivel de datos: es la que se encarga de ver quien está validado al acceso de cierta información para la consulta, actualización, borrado o inserción de datos.

La seguridad a nivel de protección: se encarga de realizar copias de los datos y también de la restauración de los mismos. [20]

Capítulo 3. Materiales y Métodos

En este capítulo se explica el proceso para la realización de la aplicación móvil y de los programas que conforman el proyecto en su totalidad.

3.1 Descripción del área de estudio

El área de estudio de este proyecto abarca el desarrollo de aplicaciones para Android y el concepto de web services. La aplicación móvil obtendrá después de la invocación de un web method, los datos que solicite y los presentará en pantalla al usuario. También se tratan a las aplicaciones que residen en el servidor y que recolectan los datos que son devueltos por el web service a la aplicación móvil.

3.2 Materiales

Para el desarrollo de la aplicación y su prueba se utilizaron los siguientes materiales:

- Celular smartphone LG modelo L7 P700 con sistema operativo Android.



Figura 3.1. Móvil LG L7.

- Computadora para el desarrollo de software, Samsung R480 con sistema operativo Windows 7, con una memoria RAM de 3GB y un disco duro de 280GB.



Figura 3.2. Laptop Samsung R480.

- Una computadora designada como servidor Gateway NV55S con sistema operativo Windows 7, con una memoria RAM de 4GB y un disco duro de 280GB.



Figura 3.3. Laptop Gateway NV55S.

Y para el desarrollo de las aplicaciones se utilizaron las siguientes herramientas:

- Eclipse Juno Service Release 1
- Android Development Toolkit Version 20.0.3
- Visual Studio 2010 Ultimate Version 10.0
- Microsoft .NET Framework Version 4.5
- SQL Server 2008 Version 10.0

3.3 Métodos

Al iniciar el proyecto se tomó la decisión de realizar la aplicación móvil para la plataforma de Google Android debido a que el uso de este sistema operativo no se restringe a los teléfonos de marca Apple sino que es instalado en una gran mayoría de marcas y sus desarrollos y publicación no se limitaban a computadoras Apple, sino que las aplicaciones para Android pueden ser desarrolladas en una gran variedad de marcas de computadoras que se ofrecen en el mercado; de igual manera se consideró desde el

inicio que la aplicación móvil debía comunicarse a un servidor, la base de datos que residía en el servidor debía estar trabajando en Microsoft SQL Server debido a que este es el gestor de base de datos que tradicionalmente encontramos en aplicaciones empresariales.

Se omitirá la instalación de los paquetes utilizados como Eclipse, SQL Server 2008, y Visual Studio 2010 ya que no se deben hacer ajustes especiales de las instalaciones típicas para desarrollar este proyecto.

En la Figura 3.4 se presenta gráficamente la idea del modelo cliente servidor que se emplea en el desarrollo del presente proyecto.

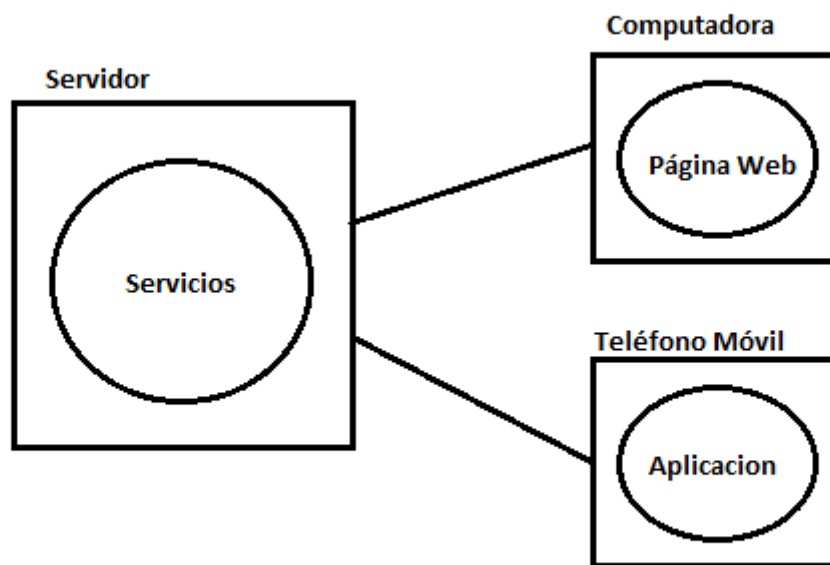


Figura 3.4: Mapa de desarrollo.

Antes de comenzar con la programación de las aplicaciones de la página web y del teléfono móvil, se definió que es lo que debía proporcionárseles a dichas aplicaciones desde el servidor, así que los servicios fueron la primera parte que se decidió programar, dichos servicios se detallan a continuación.

3.3.1 Web service para Android

En este web service se incluyen los web method que son invocados por la aplicación móvil y cuyos resultados son interpretados y mostrados directamente al usuario.

Este es el web method que sirve para obtener los valores de los tipos de cambio de peso - dólar almacenados en la base de datos y que posteriormente interpretará la aplicación para el usuario:

```
[WebMethod] ////////////////////////////////////////////////// casa de cambio "valor"

    public string valor(string iniciador)
    {
        try
        {
            // Creando objetos de datos.
            System.Data.DataSet ds = new System.Data.DataSet();
            System.Data.SqlClient.SqlDataAdapter da = new
            System.Data.SqlClient.SqlDataAdapter("SELECT compra, venta FROM Usuarios WHERE
            casadecambio = @casadecambio ", " Server=198.216.59.184;User Id =prueba;
            password=****; Initial Catalog=clientes;");

            // Añadiendo el parámetro
            da.SelectCommand.Parameters.Add("@casadecambio",
            System.Data.SqlDbType.VarChar);

            da.SelectCommand.Parameters["@casadecambio"].Value = iniciador;

            // Se recuperan los datos
            da.Fill(ds, "usuarios");

            // Se verifica la existencia del registro
            if (ds.Tables["usuarios"].Rows.Count != 0)
            {
                string cantidadcompra;
                string cantidadventa;
                string cantidadtotal;

                cantidadcompra =
                ds.Tables["usuarios"].Rows[0]["compra"].ToString();
                cantidadventa =
                ds.Tables["usuarios"].Rows[0]["venta"].ToString();

                cantidadtotal = string.Concat("Compra: $", cantidadcompra, "
                Venta: $", cantidadventa);
                return cantidadtotal;
            }
        }
    }
}
```

```

    }
    else
    {
        string resultado3 = "ERROR";
        return resultado3;
    }
}

```

Como se puede observar la tabla usuarios contiene los valores de compra y venta de la casa de cambio y es enviado como respuesta del web method “valor”.

Para código completo ver anexo 2.3.

El siguiente web method obtiene el tiempo de espera para vehículos de pasajeros en el cruce internacional de puente de las américas en Ciudad Juárez:

[WebMethod] //////////////juárez américas

```

public string EstadoDePuentes()
{
    string iniciador = "americas";
    if (iniciador == "americas")
    {
        string a = "No Existe Conexión al Puente";

        int i = 0;

        XmlTextReader reader = new
        XmlTextReader("http://apps.cbp.gov/bwt/bwt.xml");
        XmlNodeType type;
        while (reader.Read())
        {
            type = reader.NodeType;

            if (type == XmlNodeType.Element)
            {
                if (reader.Name == "port_number")
                {
                    reader.Read();

                    if (reader.Value == "240201")
                    {
                        while (reader.Read())

                            if (reader.Name == "passenger_vehicle_lanes")
                            {

```

```

        while (reader.Read())
        {
            if (reader.Name == "delay_minutes")
            {
                reader.Read();
                i++;
                if (i == 1)
                {
                    a = reader.Value;
                    reader.Close();
                } } } } }
        reader.Close();

        if (a == "")
        {
            return ("No Hay Conexion con el Puente");
        }
        else return a;
    }
}

```

En el código anterior se puede observar que el resultado del web method es el tiempo de espera para la línea de vehículos de pasajeros para el puente 240201 que es el número de puerto de entrada a Estados Unidos asignado en la página web oficial de U.S. Custom and Border Protection para el puente de las américas o puente libre en Cd. Juárez.

Para código completo ver anexo 2.3.

El siguiente web method regresa el clima actual para la Ciudad Seleccionada en la aplicación móvil:

[WebMethod] ///// Clima en Cd. Juárez///

```

public string climaenjuarez()
{
    string iniciador = "juarez";

    if (iniciador == "juarez")
    {
        string a = "No Existe Conexión al servidor";

        XmlTextReader reader = new
        XmlTextReader("http://espanol.weather.com/rss-Ciudad-Juarez-MXCA0026");
    }
}

```

```

        XmlNodeType type;
while (reader.Read())
{
    type = reader.NodeType;

    if (type == XmlNodeType.Element)
    {
        if (reader.Name == "item")
        {
            while (reader.Read())

                if (reader.Name == "description")
                {
                    reader.Read();

                    a = reader.Value;

                    a = a.Replace("&deg;", "°");
                    a =
a.Replace("Para obtener más
detalles...", " ");
                    reader.Close();

                }
            }
        }
    }
}

reader.Close();
return a;
}

return ("no existe conexion");
}

```

Aquí se obtiene el resultado como una cadena de caracteres en donde se sustituye ° por el símbolo de grados (°) y eliminar “Para obtener más detalles...” de la cadena. Enviando el resultado como se desea que aparezca en la aplicación (20° por ejemplo).

Para código completo ver anexo 2.3.

3.3.2 Servicio para página web

La página web que puede ser vista por los usuarios desde una computadora hace uso básicamente de los mismos métodos del web service que la aplicación móvil, con la diferencia de que en la página web los representantes de las casas de cambio pueden

acceder a través de un usuario y contraseña para modificar el tipo de cambio que es ofertado en su negocio, la porción de código del web method que se utiliza para tal efecto es el siguiente:

```
[WebMethod] // Guardar Cambios dependiendo de la Casa de Cambio

public string guardarcambio(string compra, string venta, string casa)
{
    string resultado="ERROR";
    string resultado2="OK";

    try
    {

        System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184;User Id =prueba;
password=****; Initial Catalog=clientes;");
        System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

        // Estableciendo propiedades
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE usuarios SET compra = @compra , venta =
@venta WHERE casadecambio = @casadecambio ";

        cmd.Parameters.Add("@casadecambio",
System.Data.SqlDbType.VarChar);
        cmd.Parameters.Add("@compra", System.Data.SqlDbType.VarChar);
        cmd.Parameters.Add("@venta", System.Data.SqlDbType.VarChar);

        cmd.Parameters["@casadecambio"].Value = casa;
        cmd.Parameters["@compra"].Value = compra;
        cmd.Parameters["@venta"].Value = venta;

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        return resultado2;

    }

    catch (System.Exception ex)
    {
        return resultado;
    }
}
```

Ver código completo en anexo 3.2.

El resultado de este web method es la actualización en la tabla usuarios de los registros de compra y venta para una casa de cambio en específico. Para que los nuevos valores sean mostrados en la aplicación móvil como en la página web.

3.3.3 Captura de tiempos para estadísticas

En esta sección se detalla la aplicación que obtiene los tiempos de cruce y los guarda en una tabla de la base de datos, para que sean mostrados según la hora de captura, con el fin de que el usuario pueda hacer una estimación del posible tiempo de cruce que encontrará posteriormente en el día, según el tiempo capturado el mismo día de la semana pasada.

Básicamente consiste en una aplicación que está en ejecución constante, determinando si la hora de captura está dentro del intervalo de 0 a 25 minutos después de las horas que serán mostradas en la aplicación en la pantalla de estadísticas, es decir 7am, 9am, 11am, 1pm, 3pm, 5pm, 7pm y 9pm.

```
string horadeldia = "";
string diadelasemana = "";

if (dt.Hour == 7 && dt.Minute <= 25)
{
    horadeldia = "sieteam";
}
```

Ver código completo en anexo 3.3

Para actualizar la tabla en la base de datos con el tiempo que se está capturando se utiliza este código:

```
if (horadeldia != "")
{
    try
    {
        System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184; User Id =prueba;
password=****; Initial Catalog=clientes;");
        System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

        // Estableciendo propiedades
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasisidro SET " + horadeldia + " = @hora WHERE
dia = '" + diadelasemana + "' ";
        cmd.Parameters.Add("@hora", System.Data.SqlDbType.Text);
        cmd.Parameters["@hora"].Value = tiempo.ToString();
        conn.Open();
        cmd.ExecuteNonQuery();
    }
}
```

```
conn.Close();
```

Ver código completo en anexo 3.3.

Esta misma lógica es empleada para guardar los valores de tiempo de los demás puentes internacionales en sus respectivas tablas y en las columnas que indiquen las horas de captura.

3.3.4 Aplicación móvil

Aquí se muestra el layout principal de la aplicación móvil:

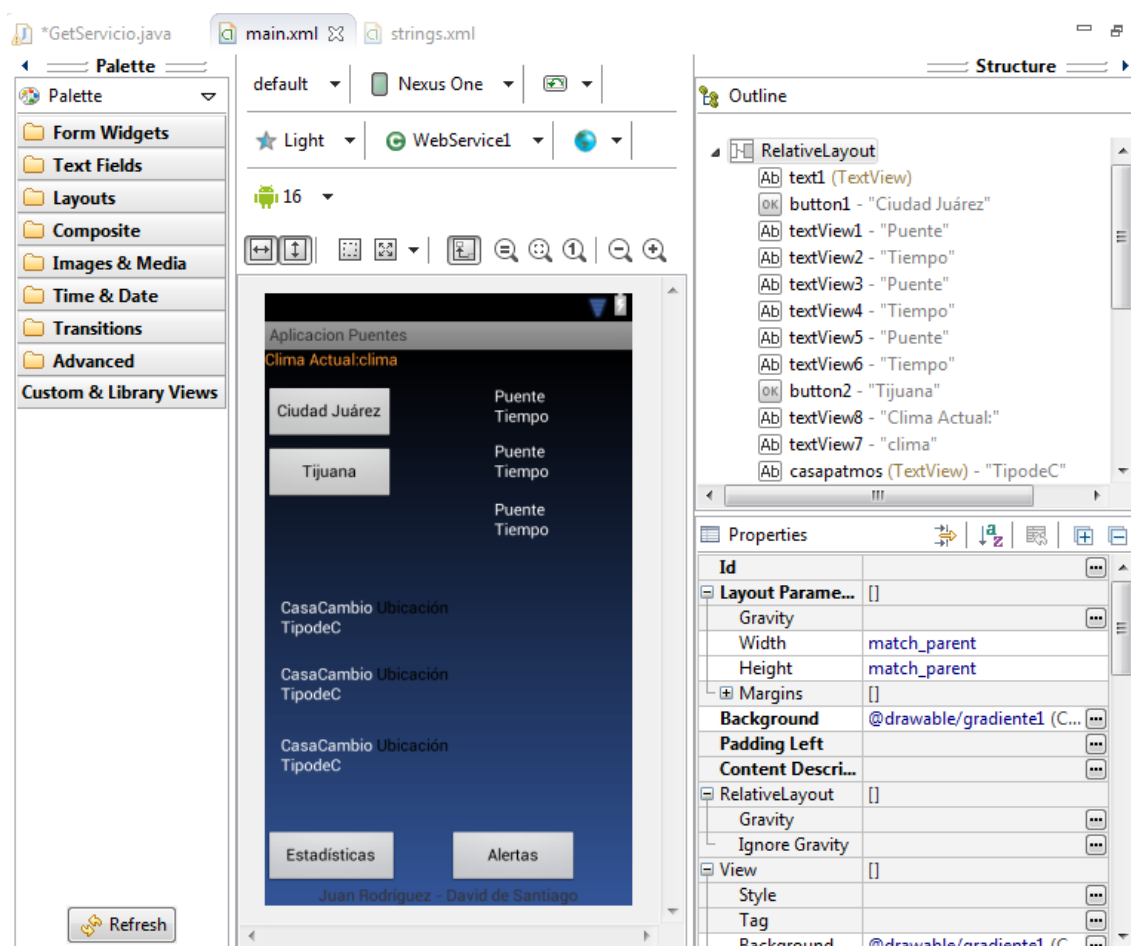


Figura 3.5: Layout de la aplicación móvil.

El código que conforma a la actividad principal se muestra a continuación:

En esta parte del código se muestran los métodos a los cuales accederá la aplicación a través del web service llamado service1.asmx y cuyos parámetros así como resultados serán manejados por el API de SOAP llamado KSOAP.

```
private static final String accionSoap =
"http://tempuri.org/EstadoDePuentes";
    private static final String accionSoap2 =
"http://tempuri.org/EstadoDePuentes2";
    private static final String accionSoap3 =
"http://tempuri.org/EstadoDePuentes3";
    private static final String accionSoap4 =
"http://tempuri.org/EstadoDePuentes4";
    private static final String accionSoap5 =
"http://tempuri.org/climaenjuarez";
    private static final String accionSoap6 =
"http://tempuri.org/climaentijuana";
    private static final String accionSoap7 =
"http://tempuri.org/valor";

    private static final String Metodo = "EstadoDePuentes";
    private static final String Metodo2 = "EstadoDePuentes2";
    private static final String Metodo3= "EstadoDePuentes3";
    private static final String Metodo4= "EstadoDePuentes4";
    private static final String Metodo5= "climaenjuarez";
    private static final String Metodo6= "climaentijuana";
    private static final String Metodo7= "valor";

    private static final String namespace = "http://tempuri.org/";
    private static final String url =
"http://198.216.59.184/Servicio%20Android/service1.asmx";
```

Ver código completo en anexo 3.4.

A continuación se detalla el proceso para obtener el tipo de cambio de las casas de cambio participantes:

```
try { //Casas de Cambio

    // request
    SoapObject request7 = new SoapObject(namespace,
Metodo7);
    SoapObject request8 = new SoapObject(namespace,
Metodo7);
    SoapObject request9 = new SoapObject(namespace,
Metodo7);

    //enviar parametros al metodo
    request7.addProperty("iniciador", "patmos");
    request8.addProperty("iniciador", "valor");
```



```

request9.addProperty("iniciador", "mr. dolar");

// modelo del sobre
SoapSerializationEnvelope sobre7 = new
SoapSerializationEnvelope(
    SoapEnvelope.VER11);
sobre7.dotNet = true;
sobre7.setOutputSoapObject(request7);

SoapSerializationEnvelope sobre8 = new
SoapSerializationEnvelope(
    SoapEnvelope.VER11);
sobre8.dotNet = true;
sobre8.setOutputSoapObject(request8);

SoapSerializationEnvelope sobre9 = new
SoapSerializationEnvelope(
    SoapEnvelope.VER11);
sobre9.dotNet = true;
sobre9.setOutputSoapObject(request9);

// modelo de transporte
HttpTransportSE transporte7 = new
HttpTransportSE(url);
HttpTransportSE transporte8 = new
HttpTransportSE(url);
HttpTransportSE transporte9 = new

// llamada
transporte7.call(accionSoap7, sobre7);
transporte8.call(accionSoap7, sobre8);
transporte9.call(accionSoap7, sobre9);

// Resultado
SoapPrimitive resultado7 = (SoapPrimitive)
sobre7.getResponse();
SoapPrimitive resultado8 = (SoapPrimitive)
sobre8.getResponse();
SoapPrimitive resultado9 = (SoapPrimitive)
sobre9.getResponse();

texto8.setText("" + resultado7.toString());
texto9.setText("Patmos:");

texto10.setText("" + resultado8.toString());
texto11.setText("Valor:");

texto13.setText("" + resultado9.toString());
texto12.setText("Mr. Dollar:");

```

```

    }

    catch (Exception e) {
        tv.setText(e.getMessage());
    }

```

Ver código completo en anexo 3.4

Como se puede observar para consumir un servicio web con el protocolo SOAP es necesario llevar a cabo 5 pasos, definir la petición o request, configurar el sobre o envelope, definir el modelo de transporte, hacer la llamada o call, y recibir el resultado.

La siguiente porción de código sirve para invocar al web method que proporciona los tiempos de cruce así como recibir los resultados:

```

try {/////el puente Américas en Juárez

    // request
    SoapObject request = new SoapObject(namespace,
Metodo);

    // modelo del sobre
    SoapSerializationEnvelope sobre = new
SoapSerializationEnvelope(
        SoapEnvelope.VER11);
    sobre.dotNet = true;
    sobre.setOutputSoapObject(request);

    // modelo de transporte
    HttpTransportSE transporte = new
HttpTransportSE(url);

    // llamada
    transporte.call(accionSoap, sobre);

    // Resultado
    SoapPrimitive resultado1 = (SoapPrimitive)
sobre.getResponse();

    texto1.setText("Américas:");
    texto2.setText("" + resultado1.toString());

}

catch (Exception e) {
    tv.setText(e.getMessage());
}

```

}

Ver código completo en anexo 3.4.

Como se puede observar el proceso es muy similar al de la obtención de los tipos de cambio mostrada previamente.

En el anexo 3.5 se muestra el código que genera la base de datos para poder comprobar las aplicaciones anteriores.

3.4 Definición de la métrica de evaluación de resultados de la aplicación

La métrica que se estableció se divide en dos partes consistente en:

- La velocidad con que la aplicación móvil muestra la información.
- La veracidad de los datos mostrados al usuario.

La velocidad con que presenta la aplicación móvil los datos al usuario depende de varios factores, entre los que se encuentra el procesador del teléfono móvil del usuario, la velocidad de transferencia de datos de la red a la que se encuentre conectado en ese momento, de la velocidad de transferencia de datos de la red del servidor y de la velocidad de procesamiento de las peticiones que hace la aplicación móvil al servidor, es decir, la velocidad de procesamiento del servidor. Estos datos son recogidos en la Tabla 4.1 que se encuentran en el capítulo 4 de Resultados de la Investigación.

Es de esta manera como se concluye con el desarrollo de la aplicación móvil, la programación de los web services así como de la programación de las aplicaciones que recopilan datos para después ser mostrados en la aplicación móvil. Sin embargo no se concluye con el mejoramiento del software que conforma el proyecto, ya que sigue siendo ampliamente perfectible en todas sus secciones.

Capítulo 4. Resultados de la Investigación

El proyecto Aplicación Móvil Para Visualización de Tiempos de Cruce en Puentes Internacionales, Clima y Tipo de Cambio en Ciudades Fronterizas, obtuvo como resultado una aplicación para la plataforma Android de fácil uso, que concentra los tiempos de espera para los cruces internacionales de las ciudades fronterizas de Tijuana Baja California, y Juárez, Chihuahua, muestra el tiempo de espera registrado la semana anterior, muestra la evolución por hora del clima en la ciudad seleccionada, muestra los tipos de cambio ofertados para las casas de cambio participantes y ofrece la programación de alarmas para avisar de tiempos de cruce deseados.

4.1 Presentación de los resultados

La siguiente tabla contiene los resultados de medir el tiempo que tarda en presentar los datos al usuario la aplicación móvil, en las diferentes pantallas que la componen:

Pantalla	Tiempo para presentar datos en pantalla	Datos transferidos	Hora del día	Tipo de conexión		
Tiempos de espera	3.708 seg	8 kb	19:16	WIFI	2.68/0.31 Mbps	19.36/1.91 Mbps
Tiempos de espera	7.513 seg	8 kb	19:11	3g/telcel	1.66/0.13 Mbps	19.36/1.91 Mbps
Estadísticas	1.927 seg	3 kb	19:16	WIFI	2.68/0.31 Mbps	19.36/1.91 Mbps
Estadísticas	3.425 seg	3 kb	19:12	3g/telcel	1.66/0.13 Mbps	19.36/1.91 Mbps
Alertas	0	14 kb	19:16	WIFI	2.68/0.31 Mbps	19.36/1.91 Mbps
Clima por hora	1.932 seg	52kb	19:17	WIFI	2.68/0.31 Mbps	19.36/1.91 Mbps
Clima por hora	2.441 seg	52kb	19:13	3g/telcel	1.66/0.13 Mbps	19.36/1.91 Mbps

Tabla 4.1: Tabla de medición de tiempo de pantallas de aplicación móvil

La siguiente tabla muestra la diferencia entre los datos mostrados por algunas fuentes de información con respecto a la aplicación móvil.

Fuente	Puente internacional	Tiempo de espera	Clima Cd. Juárez Chih.	Clima Tijuana B.C.	Hora de consulta
www.reportedepuentes.com	Américas	38 min			21:00
http://reportegaritas.com/cdjuarez	Américas	40 min			21:00
http://diario.mx/menu/reporte_puentes.php	Américas	35 min			21:00
Aplicación de Tiempos de Cruce	Américas	38 min			21:00
www.reportedepuentes.com	Santa Fe	26 min			21:10
http://reportegaritas.com/cdjuarez	Santa Fe	45 min			21:10
http://diario.mx/menu/reporte_puentes.php	Santa Fe	26 min			21:10
Aplicación de Tiempos de Cruce	Santa Fe	26 min			21:10
http://garitacenter.com/	San Ysidro	30 min			21:16
http://www.garitas.mx/Tijuana	San Ysidro	30 min			21:16
Aplicación de Tiempos de Cruce	San Ysidro	30 min			21:16
www.diario.mx			26° C		21:27
www.weather.com			27° C		21:27
http://www2.uacj.mx/clima/			26.6° C		21:27
Aplicación de Tiempos de Cruce			27° C		21:27
www.weather.com				16° C	21:30
http://www.accuweather.com				16° C	21:30
http://www.frontera.info/Clima/				16° C	21:30
Aplicación de Tiempos de Cruce				16° C	21:30

Tabla 4.2: Tabla de diferencia entre diferentes fuentes.

La aplicación consta de 4 pantallas diferentes que son: tiempos de espera, estadísticas, alertas y clima por hora.

La pantalla de Tiempos de espera muestra los botones para selección de la ciudad a mostrar, una vez seleccionada la ciudad se despliega el tiempo de espera en los cruces internacionales, el clima actual y los tipos de cambio ofertados por las casas de cambio que participen en el proyecto.



Figura 4.1: Pantalla principal de la aplicación.

Cada casa de cambio tiene una liga que al ser seleccionada conduce a su ubicación en un mapa de la ciudad.

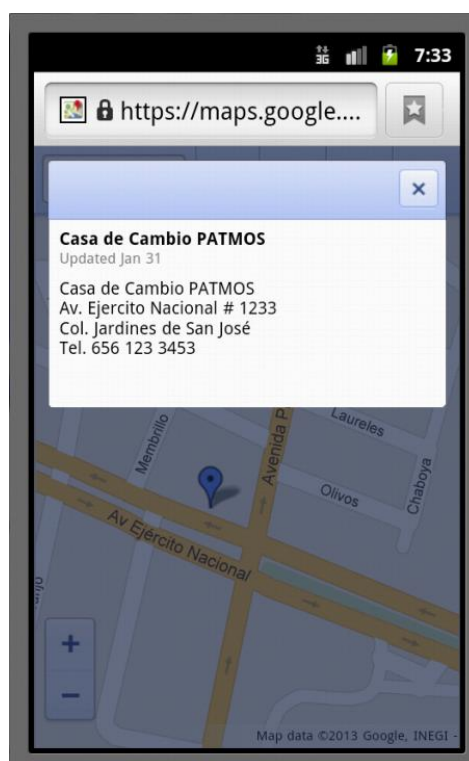


Figura 4.2: Pantalla que muestra la ubicación de las casas de cambio.

La pantalla de estadísticas muestra los registros de tiempo capturados la semana anterior, esto con la finalidad de que el usuario pueda observar el comportamiento de los tiempos de espera para los puentes de la ciudad seleccionada y tomar una decisión en cuanto al momento del día y al cruce que representan la mejor opción para el usuario. También la pantalla de estadísticas el tiempo actual en cada cruce y su hora de captura.

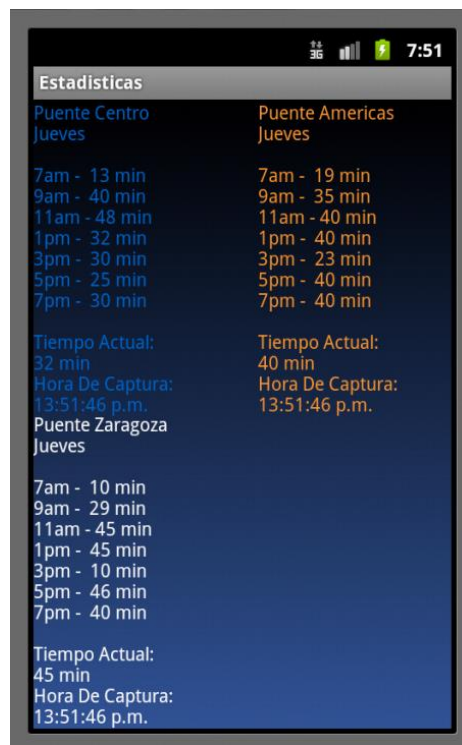


Figura 4.3: Pantalla que muestra las estadísticas de los tiempos de cruce de cada puente.

La pantalla de alertas permite establecer una alarma para que sea activada cuando el tiempo de cruce este por debajo de la cantidad de tiempo en minutos seleccionada, es decir si es seleccionado “20 minutos” para el puente Américas, entonces la alarma se activará cuando el tiempo de cruce este en 20 minutos o debajo de esa cantidad de tiempo.



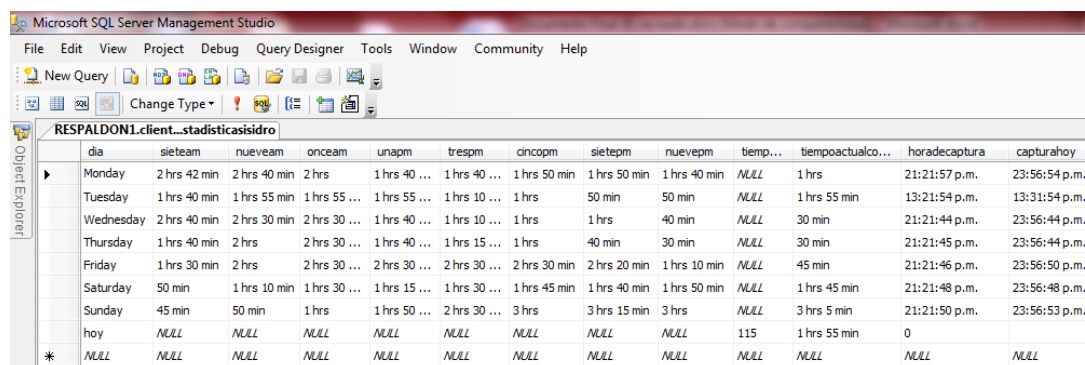
Figura 4.4: Pantalla que muestra la sección de alertas para los tiempos de los puentes.

La pantalla de clima por hora muestra el clima para las próximas 5 horas en una imagen compuesta por la hora, el clima previsto, la sensación térmica, la humedad, la probabilidad de lluvia y la velocidad del viento así como su dirección, esta información puede ser útil para tomar una decisión en cuanto al momento de cruzar por el puente internacional, ya que el clima puede ser un factor determinante para algunas personas que utilizan los cruces internacionales.



Figura 4.5: Pantalla que muestra el clima por hora de la ciudad.

En las tablas de la base de datos se concentran los tiempos capturados a las diferentes horas del día que son mostrados en la pantalla de estadísticas de tiempo, existe una tabla diferente para cada cruce internacional y a continuación se muestra la del cruce San Ysidro en Tijuana, Baja California.



dia	sieteam	nueveam	onceam	unapm	trespm	cincopm	sietepm	nuevepm	tiemp...	tiempoactualco...	horadecaptura	capturahoy
Monday	2 hrs 42 min	2 hrs 40 min	2 hrs	1 hrs 40 ...	1 hrs 40 ...	1 hrs 50 min	1 hrs 50 min	1 hrs 40 min	NULL	1 hrs	21:21:57 p.m.	23:56:54 p.m.
Tuesday	1 hrs 40 min	1 hrs 55 min	1 hrs 55 ...	1 hrs 55 ...	1 hrs 10 ...	1 hrs	50 min	50 min	NULL	1 hrs 55 min	13:21:54 p.m.	13:31:54 p.m.
Wednesday	2 hrs 40 min	2 hrs 30 min	2 hrs 30 ...	1 hrs 40 ...	1 hrs 10 ...	1 hrs	1 hrs	40 min	NULL	30 min	21:21:44 p.m.	23:56:44 p.m.
Thursday	1 hrs 40 min	2 hrs	2 hrs 30 ...	1 hrs 40 ...	1 hrs 15 ...	1 hrs	40 min	30 min	NULL	30 min	21:21:45 p.m.	23:56:44 p.m.
Friday	1 hrs 30 min	2 hrs	2 hrs 30 ...	2 hrs 30 ...	2 hrs 30 ...	2 hrs 30 min	2 hrs 20 min	1 hrs 10 min	NULL	45 min	21:21:46 p.m.	23:56:50 p.m.
Saturday	50 min	1 hrs 10 min	1 hrs 30 ...	1 hrs 15 ...	1 hrs 30 ...	1 hrs 45 min	1 hrs 40 min	1 hrs 50 min	NULL	1 hrs 45 min	21:21:48 p.m.	23:56:48 p.m.
Sunday	45 min	50 min	1 hrs	1 hrs 50 ...	2 hrs 30 ...	3 hrs	3 hrs 15 min	3 hrs	NULL	3 hrs 5 min	21:21:50 p.m.	23:56:53 p.m.
hoy	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	115	1 hrs 55 min	0	
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tabla 4.3: Tabla de registros de tiempo para cruce San Ysidro

El análisis de la información de la tabla 4.3 se incluye en la sección análisis e interpretación de resultados.

4.2 Análisis e interpretación de resultados

Analizando el resultado de la pantalla de tiempos de espera se puede observar que la base de datos funciona correctamente al mostrar los tipos de cambio almacenados así como el tiempo de espera en los puentes internacionales, ya que también se encuentran almacenados en la base de datos que reside en el servidor.

El clima que se muestra en la pantalla de tiempos de espera no es almacenado en el servidor, el web service obtiene directamente de la fuente esa información asegurando así que el clima no tenga ningún retraso, ya la obtención del clima representa muy poco esfuerzo para el servidor esto se traduce en un solo un retraso de para mostrar los datos al usuario de la aplicación móvil.

La ubicación de las casas de cambio en un mapa son mostradas con la ayuda de del web browser, o con la aplicación por defecto de mapas del móvil. Esto representa un tiempo de espera que está sujeto a la velocidad de la red y a la velocidad del procesador del móvil que ejecuta dicha aplicación.

La pantalla de estadísticas muestra los registros capturados cada 2 horas desde las 7:00 hrs. hasta las 19:00 hrs. del día en el cuál se hace la consulta, es decir, si hoy fuera viernes, mostraría los registros capturados el viernes anterior esto con la finalidad de que el usuario pueda observar las tendencias de tiempo de espera en el día y ayudarlo a decidir la hora para arribar al cruce fronterizo. Esta pantalla es llenada con la información que reside en la base de datos misma que es actualizada por el programa de captura de tiempos para estadísticas.

El análisis de la pantalla de alertas resulta en una funcionalidad parecida a la alarma de un despertador, en éste cuando el contador llega a cero la alarma se activa indicando que el tiempo ha llegado al establecido por el usuario. De igual manera en la pantalla de alertas la alarma se activa cuando el tiempo de cruce ha llegado a ser igual o inferior al tiempo de espera establecido por el usuario mostrando una pantalla de alerta con el tiempo de espera y el cruce señalados. Los tiempos de espera configurables son de 20, 40, 60 y 120 minutos.

En cuanto a la pantalla de clima por hora, en ella se muestra una imagen que tiene 7 divisiones, la hora, un icono que representa el estado general del clima (soleado, nublado, con viento, con lluvia, con nieve, etcétera), la predicción del clima para esa hora del día, la sensación térmica o sensación aparente de la temperatura por las personas, la humedad en el ambiente, la probabilidad de precipitación y por último, la velocidad y dirección del viento. Estos datos son tomados directamente de la página de www.weather.com y son actualizados cada 5 minutos en la base de datos.

La tabla de la base de datos que muestra los registros de tiempo capturados para cada cruce en específico muestra una columna para los días de la semana, incluyendo una tupla para el día “hoy”, 8 columnas para horas del día, dos columnas para último tiempo capturado y una columna para la hora de captura del último tiempo capturado.

La columna “día”, contiene los días de la semana, cada columna posterior incluye el tiempo de espera registrado en esa hora y día específicamente y es mostrada en la pantalla de estadísticas dependiendo del día de consulta, es decir, si hoy fuera lunes, mostraría los tiempos registrados para cada hora de las 9am a las 7pm del día lunes anterior. La dos columnas de “tiempo” muestran una el tiempo en horas y minutos para ser desplegado en la pantalla de consulta de tiempos de cruce, y la otra contiene el tiempo solo en minutos que es utilizado para las consultas que realiza la pantalla de alertas en cuanto a tiempos de cruce en específico. La columna “capturahoy” muestra la

hora de captura que es posteriormente desplegada en la pantalla de estadísticas con el fin de comprobar que tan actualizado está el registro de tiempo de cruce que se muestra al usuario.

Capítulo 5. Discusiones, conclusiones y recomendaciones

La problemática que se intenta solucionar al inicio del proyecto es la de ofrecer una herramienta que concentre tiempo de cruce, clima de la ciudad y tipo de cambio ofertado en casas de cambio locales, esta herramienta está pensada para brindar esta información al usuario sin la necesidad de que consulte a través de diferentes medios o fuentes de información esos datos, se añadieron algunas funcionalidades que pueden resultarle útiles, como alertas para tiempos específicos de cruce y registros de tiempos de cruce en diferentes puentes internacionales.

Esta herramienta fue desarrollada para ejecutarse en la plataforma móvil Android y utiliza una base de datos para contener la información que es desplegada al usuario en las diferentes pantallas que conforman a la aplicación.

También desarrolló la página de Internet www.tiempoenpuentes.com que muestra el tipo de cambio ofertado en las casas de cambio, brinda la opción para que los representantes de cada negociación modifiquen el tipo de cambio que ofertan mismo que es visualizado de igual manera en la aplicación móvil. También es desplegado el tiempo de cruce así como el clima que aparece en la aplicación móvil.

5.1 Con respecto a las preguntas de investigación

Los obstáculos que se presentaron para la obtención y manejo de los datos fueron los propios del hecho de trabajar con herramientas que no habíamos usado antes como lo fue Eclipse para Android y el lenguaje de programación Java, la mayor parte de los datos obtenidos de las fuentes de información son xml y los datos transferidos de la aplicación móvil al servidor son a través del protocolo SOAP que también utiliza xml para tal efecto.

Solo existe una fuente de información oficial para el tiempo de cruce de los puentes internacionales y es la que aparece en la página de U.S. Customs and Border Protection y que contiene información de espera de los puertos de entrada a Estados Unidos.

En cuanto al clima existen diversas fuentes fiables, mexicanas y extranjeras, sin embargo para este proyecto se utilizó la fuente de www.weather.com debido a que

permite y brinda el uso de código para utilizarlo en páginas externas, además de que The Weather Channel propietario de la página de internet es uno de los pronosticadores de clima más famosos en el mundo.

Las fuentes de tipo de cambio en casas de cambio en Ciudad Juárez actualmente son solo los establecimientos propios, aún no existe una fuente publicada en una página de internet que brinde esa información al público en general. Sin embargo el presente proyecto en sus trabajos futuros ofrece una alternativa para esta situación al ofrecer un espacio gratuito en internet para ofrecer sus servicios.

5.2 Con respecto al objetivo de la investigación

Se logró obtener una aplicación que cumple con el objetivo al mostrar en la aplicación móvil los tiempos de cruce de los puentes internacionales, el clima de la ciudad y el tipo de cambio de las casas de cambio participantes.

Y se desarrolló la página web www.tiempoenpuentes.com que muestra la información de clima actual, tiempos de cruce y tipos de cambio que aparecen en la aplicación móvil.

5.3 Recomendaciones para futuras investigaciones

Investigar y desarrollar una aplicación para el sistema operativo IOS de Apple ya que es una plataforma que abarca gran parte de la cuota de mercado de teléfonos inteligentes o smart phones vendidos en el mundo.

Invitar a las casa de cambio a participar en el proyecto con actualizaciones de los tipos de cambio que ofrecen, interesarlos en base a un espacio publicitario gratuito en internet y que los usuarios de la página web tengan a la mano información actualizada por la misma casa de cambio.

También otro paso es el de programar en la aplicación un espacio de interacción entre los usuarios de manera que puedan ellos mismos hacer comentarios en cuanto al tiempo de cruce que experimentaron en el puente fronterizo que utilizaron, así como de sugerencias para la mejora de la aplicación.

Tener presencia en redes sociales con actualizaciones de los tiempos de cruce y clima en una publicación que tenga ligas de descarga al mercado de aplicaciones de la plataforma que utilice el usuario.

También es importante en una futura modificación el incluir una forma de financiar la investigación, desarrollo e implementación de la aplicación, así como la adquisición de un mejor y más potente servidor, con una conexión de internet más rápida, todo esto a través de incluir publicidad en la página web o en la aplicación para costear de dichas mejoras.

Referencias

- [1] Sitio web de la ciudad de El Paso, Tx, “The City of El Paso International Bridges”, Abril 2013, <http://home.elpasotexas.gov/bridges/traffic-volume.php>
- [2] Securing America’s Borders “U.S. Customs and Border Protection” Department of Homeland Security, <http://www.cbp.gov/>
- [3] Aplicación para la espera de los tiempos de cruce en los puentes internacionales, “Best Time To Cross The Border”, Abril 2013, <https://play.google.com/store/apps/details?id=net.calit2.ports&hl=es>
- [4] Aplicación que trae todas las estaciones y pronósticos del tiempo de la ciudad deseada, “Yahoo! Tiempo”, Abril 2013, <https://play.google.com/store/apps/details?id=com.yahoo.mobile.client.android.weather&hl=es>
- [5] Aplicación que muestra los tipos de cambio en los diferentes bancos de Mexico, “Tipo de Cambio Tijuana” Abril 2013, <https://play.google.com/store/apps/details?id=ivano.tipoCambioTijuana>
- [6] Miguel Angel Vazquez Ruis, “Región e integración fronteriza Mexico-Estados Unidos” (1999), http://lanic.utexas.edu/project/etext/colson/17/17_4.pdf
- [7] Universidad Autónoma de Ciudad Juarez “Ciudad Juárez Chihuahua Mexico” (2013), <http://www2.uacj.mx/Universidad/Temas/visitantes/default.htm>
- [8] Jaime Aranaz Tudela, “Desarrollo de Aplicaciones Para Dispositivos Móviles Sobre la Plataforma Android de Google”, Tesis de Titulación, Universidad Carlos III de Madrid, Escuela Politécnica Superior, 2009, http://e-archivo.uc3m.es:8080/bitstream/10016/6506/1/PFC_Jaime_Aranaz_Tudela_2010116132629.pdf
- [9] .NET Web Services Solutions, Author: Jamsa, Kris Publisher: Sybex Original Publication Date: 04/2003 Subjects: Web site development. Microsoft .NET Framework.
- [10] Building Web Services with XML, Author: Arora, Geetanjali Kishore, Sai NIIT (Corporation) Staff Publisher: Premier Press, Incorporated Original Publication Date: 10/2002 Subjects: Web site development. XML (Document markup language)

- [11] Abraham Silberschatz, Henry F. Korth, Fundamentos de bases de datos, Mc GrawHill, 2002, disponible: <http://www.ceualm.com/CGI-BIN/material/20120824051303.pdf>
- [12] Manual de Referencia SQL, <http://es.scribd.com/doc/98696267/Manual-de-Referencia-SQL>
- [13] Gestores de Bases de Datos,
<http://www.di.ujen.es/~barranco/publico/ofimatica/tema7.pdf>
- [14] SQL SERVER – Descripción del Entorno y Creación de Bases de Datos,
<http://download.microsoft.com/download/f/d/3/fd3ce4c8-164b-4ec0-9a6b-d52739caaa81/ebooksql2000.pdf>
- [15] MySQL, “Que es MySQL?”,
<http://www3.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>
- [16] Jack Koftikian, “Simple Object Access Protocol (SOAP)”, <http://www.sts.tu-harburg.de/pw-and-m-theses/2001/Koft01.pdf>
- [17] Fernando Alonso Blazquez, Nicolas Serrano Barcena, Sonia Calzada Minguez, “Informática III”, “Protocolo simple de acceso a objetos (SOAP)”, San Sebastián, Febrero 2004 <http://es.scribd.com/doc/54301490/171/Protocolo-SOAP>
- [18] Diego Fernando Guamialama Nervaez, “Implementación de un sistema para monitoreo de servicios en servidores”, Escuela Politécnica Nacional, Facultad de Ingeniería Eléctrica y Electrónica, Quito, Diciembre 2008.
<http://bibdigital.epn.edu.ec/bitstream/15000/1075/1/CD-1909.pdf>
- [19] Jose Gregorio, “Servidor Web” <http://www.slideshare.net/josegregoriob/servidor-web-8451426>
- [20] Rafael Mendez, “Servidores de Bases de Datos”, Departamento Informática y Sistemas, Universidad de Murcia (2000),
<http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/sghd.html>

Anexo 2.3

Web service para aplicación móvil


```

// Se recuperan los datos
da.Fill(ds, "usuarios");

// Se verifica la existencia del registro
if (ds.Tables["usuarios"].Rows.Count != 0)
{
    string tiempo7 = "";
    string tiempo9 = "";
    string tiempo11 = "";
    string tiempo13 = "";
    string tiempo15 = "";
    string tiempo17 = "";
    string tiempo19 = "";
    string tiempo21 = "";
    string tiempoactualcompleto = "";
    string horadecaptura = "";
    // string cantidadcompra;
    // string cantidadventa;
    string cantidadtotal;

    tiempo7 =
ds.Tables["usuarios"].Rows[0]["sieteam"].ToString();
    tiempo9 =
ds.Tables["usuarios"].Rows[0]["nueveam"].ToString();
    tiempo11 =
ds.Tables["usuarios"].Rows[0]["onceam"].ToString();
    tiempo13 = ds.Tables["usuarios"].Rows[0]["unapm"].ToString();
    tiempo15 =
ds.Tables["usuarios"].Rows[0]["trespm"].ToString();
    tiempo17 =
ds.Tables["usuarios"].Rows[0]["cincopm"].ToString();
    tiempo19 =
ds.Tables["usuarios"].Rows[0]["sietepm"].ToString();
    tiempo21 =
ds.Tables["usuarios"].Rows[0]["nuevepm"].ToString();
    tiempoactualcompleto =
ds.Tables["usuarios"].Rows[0]["tiempoactualcompleto"].ToString();
    horadecaptura =
ds.Tables["usuarios"].Rows[0]["capturahoy"].ToString();

    // cantidadcompra =
ds.Tables["usuarios"].Rows[0]["compra"].ToString();
    //cantidadventa =
ds.Tables["usuarios"].Rows[0]["venta"].ToString();

    string diaespanol = "";
    if (diadelasemana == "Monday")
    { diaespanol = "Lunes"; }
    if (diadelasemana == "Tuesday")
    { diaespanol = "Martes"; }
    if (diadelasemana == "Wednesday")
    { diaespanol = "Miercoles"; }
    if (diadelasemana == "Thursday")
    { diaespanol = "Jueves"; }
    if (diadelasemana == "Friday")
    { diaespanol = "Viernes"; }
    if (diadelasemana == "Saturday")
    { diaespanol = "Sabado"; }
    if (diadelasemana == "Sunday")
    { diaespanol = "Domingo"; }
}

```

```

        cantidadtotal = string.Concat("Puente San Isidro \n",
diaespanol, "\n\n7am - ", tiempo7, "\n9am - ", tiempo9, "\n11am - ", tiempo11,
"\n1pm - ", tiempo13, "\n3pm - ", tiempo15, "\n5pm - ", tiempo17, "\n7pm - ",
tiempo19, "\n\nTiempo Actual:\n", tiempoactualcompleto, "\nHora De Captura:\n",
horadecaptura);

        return cantidadtotal;

    }
    else
    {
        string resultado3 = "ERROR";
        return resultado3;

    }

}

catch (System.Exception ex)
{
    return ex.ToString();
}

}

```

```

[WebMethod] ////////// Estadísticas del día Puente Ysleta

public string estadisticasysleta()
{
    string resultado = "";

    string diadelasemana = "";
    diadelasemana = dt.DayOfWeek.ToString();

    try
    {

        // Creando objetos de datos.
        System.Data.DataSet ds = new System.Data.DataSet();
        System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT sieteam, nueveam, onceam, unapm,
trespm, cincopm, sietepm, nuevepm, tiempoactualcompleto, capturahoy FROM
estadisticasysleta WHERE dia = @diadelasemana", " Server=198.216.59.184;User Id
=prueba; password=****; Initial Catalog=clientes;");

        // Añadiendo el parámetro
da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

```

```

da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

```

```

// Se recuperan los datos
da.Fill(ds, "usuarios");

// Se verifica la existencia del registro
if (ds.Tables["usuarios"].Rows.Count != 0)
{
    string tiempo7 = "";
    string tiempo9 = "";
    string tiempo11 = "";
    string tiempo13 = "";
    string tiempo15 = "";
    string tiempo17 = "";
    string tiempo19 = "";
    string tiempo21 = "";
    string tiempoactualcompleto = "";
    string horadecaptura = "";
    // string cantidadcompra;
    // string cantidadventa;
    string cantidadtotal;

    tiempo7 =
ds.Tables["usuarios"].Rows[0]["sieteam"].ToString();
    tiempo9 =
ds.Tables["usuarios"].Rows[0]["nueveam"].ToString();
    tiempo11 =
ds.Tables["usuarios"].Rows[0]["onceam"].ToString();
    tiempo13 = ds.Tables["usuarios"].Rows[0]["unapm"].ToString();
    tiempo15 =
ds.Tables["usuarios"].Rows[0]["trespm"].ToString();
    tiempo17 =
ds.Tables["usuarios"].Rows[0]["cincopm"].ToString();
    tiempo19 =
ds.Tables["usuarios"].Rows[0]["sietepm"].ToString();
    tiempo21 =
ds.Tables["usuarios"].Rows[0]["nuevepm"].ToString();
    tiempoactualcompleto =
ds.Tables["usuarios"].Rows[0]["tiempoactualcompleto"].ToString();
    horadecaptura =
ds.Tables["usuarios"].Rows[0]["capturahoy"].ToString();

    // cantidadcompra =
ds.Tables["usuarios"].Rows[0]["compra"].ToString();
    //cantidadventa =
ds.Tables["usuarios"].Rows[0]["venta"].ToString();

    string diaespanol = "";
    if (diadelasemana == "Monday")
    { diaespanol = "Lunes"; }
    if (diadelasemana == "Tuesday")
    { diaespanol = "Martes"; }
    if (diadelasemana == "Wednesday")
    { diaespanol = "Miercoles"; }
    if (diadelasemana == "Thursday")
    { diaespanol = "Jueves"; }
    if (diadelasemana == "Friday")

```

```

        { diaespanol = "Viernes"; }
        if (diadelasemana == "Saturday")
        { diaespanol = "Sabado"; }
        if (diadelasemana == "Sunday")
        { diaespanol = "Domingo"; }

        cantidadtotal = string.Concat("Puente Zaragoza \n",
        diaespanol, "\n\n7am - ", tiempo7, "\n9am - ", tiempo9, "\n11am - ", tiempo11,
        "\n1pm - ", tiempo13, "\n3pm - ", tiempo15, "\n5pm - ", tiempo17, "\n7pm - ",
        tiempo19, "\n\nTiempo Actual:\n", tiempoactualcompleto, "\nHora De Captura:\n",
        horadecaptura);

        return cantidadtotal;

    }
    else
    {
        string resultado3 = "ERROR";
        return resultado3;

    }

}

catch (System.Exception ex)
{
    return ex.ToString();
}

}

[WebMethod] ////////// Estadísticas del día Puente centro

public string estadisticascentro()
{
    string resultado = "";

    string diadelasemana = "";
    diadelasemana = dt.DayOfWeek.ToString();

    try
    {

        // Creando objetos de datos.
        System.Data.DataSet ds = new System.Data.DataSet();
        System.Data.SqlClient.SqlDataAdapter da = new
        System.Data.SqlClient.SqlDataAdapter("SELECT sieteam, nueveam, onceam, unapm,
        trespm, cincopm, sietepm, nuevepm, tiempoactualcompleto, capturahoy FROM
        estadisticascentro WHERE dia = @diadelasemana", " Server=198.216.59.184;User Id
        =prueba; password=****; Initial Catalog=clientes;");

```

```

        // Añadiendo el parámetro
        da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

        da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

        // Se recuperan los datos
        da.Fill(ds, "usuarios");

        // Se verifica la existencia del registro
        if (ds.Tables["usuarios"].Rows.Count != 0)
        {
            string tiempo7 = "";
            string tiempo9 = "";
            string tiempo11 = "";
            string tiempo13 = "";
            string tiempo15 = "";
            string tiempo17 = "";
            string tiempo19 = "";
            string tiempo21 = "";
            string tiempoactualcompleto = "";
            string horadecaptura = "";
            // string cantidadcompra;
            // string cantidadventa;
            string cantidadtotal;

            tiempo7 =
ds.Tables["usuarios"].Rows[0]["sieteam"].ToString();
            tiempo9 =
ds.Tables["usuarios"].Rows[0]["nueveam"].ToString();
            tiempo11 =
ds.Tables["usuarios"].Rows[0]["onceam"].ToString();
            tiempo13 = ds.Tables["usuarios"].Rows[0]["unapm"].ToString();
            tiempo15 =
ds.Tables["usuarios"].Rows[0]["trespm"].ToString();
            tiempo17 =
ds.Tables["usuarios"].Rows[0]["cincopm"].ToString();
            tiempo19 =
ds.Tables["usuarios"].Rows[0]["sietepm"].ToString();
            tiempo21 =
ds.Tables["usuarios"].Rows[0]["nuevepm"].ToString();
            tiempoactualcompleto =
ds.Tables["usuarios"].Rows[0]["tiempoactualcompleto"].ToString();
            horadecaptura =
ds.Tables["usuarios"].Rows[0]["capturahoy"].ToString();

            // cantidadcompra =
ds.Tables["usuarios"].Rows[0]["compra"].ToString();
            //cantidadventa =
ds.Tables["usuarios"].Rows[0]["venta"].ToString();

            string diaespanol = "";
            if (diadelasemana == "Monday")
            { diaespanol = "Lunes"; }
            if (diadelasemana == "Tuesday")
            { diaespanol = "Martes"; }
            if (diadelasemana == "Wednesday")

```

```

        { diaespanol = "Miercoles"; }
        if (diadelasemana == "Thursday")
        { diaespanol = "Jueves"; }
        if (diadelasemana == "Friday")
        { diaespanol = "Viernes"; }
        if (diadelasemana == "Saturday")
        { diaespanol = "Sabado"; }
        if (diadelasemana == "Sunday")
        { diaespanol = "Domingo"; }

        cantidadtotal = string.Concat("Puente Centro \n", diaespanol,
"\n\n7am - ", tiempo7, "\n9am - ", tiempo9, "\n11am - ", tiempo11, "\n1pm - ",
tiempo13, "\n3pm - ", tiempo15, "\n5pm - ", tiempo17, "\n7pm - ",
tiempo19, "\n\nTiempo Actual:\n", tiempoactualcompleto, "\nHora De Captura:\n",
horadecaptura);

        return cantidadtotal;

    }
    else
    {
        string resultado3 = "ERROR";
        return resultado3;

    }

}

}

catch (System.Exception ex)
{
    return ex.ToString();
}

}

[WebMethod] // Estadísticas del día Puente Libre americas

public string estadisticasamericas()
{
    string resultado = "";

    string diadelasemana = "";
    diadelasemana = dt.DayOfWeek.ToString();

    try
    {

        // Creando objetos de datos.
        System.Data.DataSet ds = new System.Data.DataSet();
        System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT sieteam, nueveam, onceam, unapm,
trespm, cincopm, sietepm, nuevepm, tiempoactualcompleto, capturahoy FROM
estadisticasamericas WHERE dia = @diadelasemana", " Server=198.216.59.184;User Id
=prueba; password=****; Initial Catalog=clientes;");

```



```

        // Añadiendo el parámetro
        da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

        da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

// Se recuperan los datos
da.Fill(ds, "usuarios");

// Se verifica la existencia del registro
if (ds.Tables["usuarios"].Rows.Count != 0)
{
    string tiempo7 = "";
    string tiempo9 = "";
    string tiempo11 = "";
    string tiempo13 = "";
    string tiempo15 = "";
    string tiempo17 = "";
    string tiempo19 = "";
    string tiempo21 = "";
    string tiempoactualcompleto = "";
    string horadecaptura = "";
    // string cantidadcompra;
    // string cantidadventa;
    string cantidadtotal;

    tiempo7 =
ds.Tables["usuarios"].Rows[0]["sieteam"].ToString();
    tiempo9 =
ds.Tables["usuarios"].Rows[0]["nueveam"].ToString();
    tiempo11 =
ds.Tables["usuarios"].Rows[0]["onceam"].ToString();
    tiempo13 = ds.Tables["usuarios"].Rows[0]["unapm"].ToString();
    tiempo15 =
ds.Tables["usuarios"].Rows[0]["trespm"].ToString();
    tiempo17 =
ds.Tables["usuarios"].Rows[0]["cincopm"].ToString();
    tiempo19 =
ds.Tables["usuarios"].Rows[0]["sietepm"].ToString();
    tiempo21 =
ds.Tables["usuarios"].Rows[0]["nuevepm"].ToString();
    tiempoactualcompleto =
ds.Tables["usuarios"].Rows[0]["tiempoactualcompleto"].ToString();
    horadecaptura =
ds.Tables["usuarios"].Rows[0]["capturahoy"].ToString();

    // cantidadcompra =
ds.Tables["usuarios"].Rows[0]["compra"].ToString();
    //cantidadventa =
ds.Tables["usuarios"].Rows[0]["venta"].ToString();

    string diaespanol = "";
    if (diadelasemana == "Monday")
    { diaespanol = "Lunes"; }
    if (diadelasemana == "Tuesday")

```

```

        { diaespanol = "Martes"; }
        if (diadelasemana == "Wednesday")
        { diaespanol = "Miercoles"; }
        if (diadelasemana == "Thursday")
        { diaespanol = "Jueves"; }
        if (diadelasemana == "Friday")
        { diaespanol = "Viernes"; }
        if (diadelasemana == "Saturday")
        { diaespanol = "Sabado"; }
        if (diadelasemana == "Sunday")
        { diaespanol = "Domingo"; }

        cantidadtotal = string.Concat("Puente Americas \n",
        diaespanol, "\n\n7am - ", tiempo7, "\n9am - ", tiempo9, "\n11am - ", tiempo11,
        "\n1pm - ", tiempo13, "\n3pm - ", tiempo15, "\n5pm - ", tiempo17, "\n7pm - ",
        tiempo19, "\n\nTiempo Actual:\n", tiempoactualcompleto, "\nHora De Captura:\n",
        horadecaptura);

        return cantidadtotal;

    }
    else
    {
        string resultado3 = "ERROR";
        return resultado3;

    }

}

catch (System.Exception ex)
{
    return ex.ToString();
}

}

[WebMethod] ////////////////////////////////////////////////// casa de cambio valor

public string valor(string iniciador)
{
    string resultado = "";

```

```

try
{
    // Creando objetos de datos.
    System.Data.DataSet ds = new System.Data.DataSet();
    System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT compra, venta FROM Usuarios WHERE
casadecambio = @casadecambio ", " Server=198.216.59.184;User Id =prueba;
password=****; Initial Catalog=clientes;");

    // Añadiendo el parámetro
da.SelectCommand.Parameters.Add("@casadecambio",
System.Data.SqlDbType.VarChar);

da.SelectCommand.Parameters["@casadecambio"].Value = iniciador;


    // Se recuperan los datos
da.Fill(ds, "usuarios");

    // Se verifica la existencia del registro
if (ds.Tables["usuarios"].Rows.Count != 0)
{
    string cantidadcompra;
    string cantidadventa;
    string cantidadtotal;

    cantidadcompra =
ds.Tables["usuarios"].Rows[0]["compra"].ToString();
    cantidadventa =
ds.Tables["usuarios"].Rows[0]["venta"].ToString();

    cantidadtotal = string.Concat("Compra: $", cantidadcompra, "
Venta: $", cantidadventa);
    return cantidadtotal;
}
else
{
    string resultado3 = "ERROR";
    return resultado3;
}

}

catch (System.Exception ex)
{
    return ex.ToString();
}

```

```
}
```

```
[WebMethod]

////////para la aplicacion android quitamos esto////////public string
EstadoDePuentes(string iniciador)

    public string EstadoDePuentes()    ///para la aplicacion web hay que
quitar esto....
    {
        string iniciador = "americas";    /// para ejemplo le ponemos de una
vez el iniciador (en android)
        string diadelasemana = "";
        diadelasemana = dt.DayOfWeek.ToString();

        if (iniciador == "americas")
        {

            try
            {

                // Creando objetos de datos.
                System.Data.DataSet ds = new System.Data.DataSet();
                System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT tiempoactualcompleto FROM
estadisticasamericas WHERE dia = @diadelasemana ", " Server=198.216.59.184;User
Id =prueba; password=****; Initial Catalog=clientes;");

                // Añadiendo el parámetro
                da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

                da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

                // Se recuperan los datos
                da.Fill(ds, "usuarios");

                // Se verifica la existencia del registro
                if (ds.Tables["usuarios"].Rows.Count != 0)
                {
                    string tiempoactual = "";
```

```

        tiempoactual =
ds.Tables["usuarios"].Rows[0]["tiempoactualcompleto"].ToString();

        return tiempoactual;

    }
    else
    {
        string resultado3 = "ERROR";
        return resultado3;

    }

}

catch (System.Exception ex)
{
    return ex.ToString();
}

}

return ("no existe conexion");
}

//se acaba el webmethod

[WebMethod]

////////para la aplicacion android quitamos esto////////public string
EstadoDePuentes(string iniciador)

    public string EstadoDePuentes2()  ////para la aplicacion web hay que
quitar esto....
    {
        string iniciador = "centro";  //// para ejemplo le ponemos de una vez
el iniciador (en android)
        string diadelasemana = "";
        diadelasemana = dt.DayOfWeek.ToString();
        if (iniciador == "centro")
        {

            try
            {

                // Creando objetos de datos.
                System.Data.DataSet ds = new System.Data.DataSet();

```

```

        System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT tiempoactualcompleto FROM
estadisticascentro WHERE dia = @diadelasemana ", " Server=198.216.59.184;User Id
=prueba; password=****; Initial Catalog=clientes;");

        // Añadiendo el parámetro
        da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

        da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

        // Se recuperan los datos
        da.Fill(ds, "usuarios");

        // Se verifica la existencia del registro
        if (ds.Tables["usuarios"].Rows.Count != 0)
        {
            string tiempoactual = "";

            tiempoactual =
ds.Tables["usuarios"].Rows[0]["tiempoactualcompleto"].ToString();

            return tiempoactual;
        }
        else
        {
            string resultado3 = "ERROR";
            return resultado3;
        }
    }

    catch (System.Exception ex)
    {
        return ex.ToString();
    }

}

return ("no existe conexion");
}

```

```

//se acaba el method
[WebMethod]

////////para la aplicacion android quitamos esto////////public string
EstadoDePuentes(string iniciador)

    public string EstadoDePuentes3()  ////para la aplicacion web hay que
quitar esto....
    {
        string iniciador = "zaragoza";  //// para ejemplo le ponemos de una
vez el iniciador (en android)
        string diadelasemana = "";
        diadelasemana = dt.DayOfWeek.ToString();
        if (iniciador == "zaragoza")
        {

            try
            {

                // Creando objetos de datos.
                System.Data.DataSet ds = new System.Data.DataSet();
                System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT tiempoactualcompleto FROM
estadisticasysleta WHERE dia = @diadelasemana ", " Server=198.216.59.184;User Id
=prueba; password=****; Initial Catalog=clientes;");

                // Añadiendo el parámetro
                da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

                da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

                // Se recuperan los datos
                da.Fill(ds, "usuarios");

                // Se verifica la existencia del registro
                if (ds.Tables["usuarios"].Rows.Count != 0)
                {
                    string tiempoactual = "";

                    tiempoactual =
ds.Tables["usuarios"].Rows[0]["tiempoactualcompleto"].ToString();

                    return tiempoactual;
                }
            }
            else

```

```

        {
            string resultado3 = "ERROR";
            return resultado3;
        }

    }

    catch (System.Exception ex)
    {
        return ex.ToString();
    }

}

return ("no existe conexion");
}

[WebMethod]

////////para la aplicacion android quitamos esto////////public string
EstadoDePuentes(string iniciador)

    public string EstadoDePuentes4()    ////para la aplicacion web hay que
quitar esto....
    {
        string iniciador = "ysidro";    //// para ejemplo le ponemos de una vez
el iniciador (en android)
        string diadelasemana = "";
        diadelasemana = dt.DayOfWeek.ToString();
        if (iniciador == "ysidro")
        {

            try
            {

                // Creando objetos de datos.
                System.Data.DataSet ds = new System.Data.DataSet();
                System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT tiempoactualcompleto FROM
estadisticasisidro WHERE dia = @diadelasemana ", " Server=198.216.59.184;User Id
=prueba; password=****; Initial Catalog=clientes;");

                // Añadiendo el parámetro

```



```

        da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

        da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

        // Se recuperan los datos
da.Fill(ds, "usuarios");

        // Se verifica la existencia del registro
if (ds.Tables["usuarios"].Rows.Count != 0)
{
    string tiempoactual = "";

    tiempoactual =
ds.Tables["usuarios"].Rows[0]["tiempoactualcompleto"].ToString();

    return tiempoactual;
}
else
{
    string resultado3 = "ERROR";
    return resultado3;
}

}

catch (System.Exception ex)
{
    return ex.ToString();
}

}

return ("no existe conexion");
}

//se acaba el webmethod

[WebMethod]

```

```

        //para la aplicacion android quitamos esto//public string
EstadoDePuentes(string iniciador)

        public string climaenjuarez()    //para la aplicacion web hay que quitar
esto....
        {
            string iniciador = "juarez";    /// para ejemplo le ponemos de una vez
el iniciador (en android)

            if (iniciador == "juarez")
            {
                string a = "No Existe Conexión al servidor";

                // int i = 0;

                XmlTextReader reader = new
XmlTextReader("http://espanol.weather.com/rss-Ciudad-Juarez-MXCA0026");
                XmlNodeType type;
                while (reader.Read())
                {
                    type = reader.NodeType;

                    if (type == XmlNodeType.Element)
                    {
                        if (reader.Name == "item")
                        {
                            while (reader.Read())

                                if (reader.Name == "description")
                                {
                                    reader.Read();

                                    a = reader.Value;

                                    a = a.Replace("°", "°");
                                    a = a.Replace("Para obtener más
detalles...", " ");

                                    reader.Close();

                                }

                            }

                        }

                    }

                }

            }

        }

```

```

    }

    // MessageBox.Show(i.ToString());
    reader.Close();

    return a;
}

return ("no existe conexion");
}

//////////acaba el webmethod

[WebMethod]

//////////para la aplicacion android quitamos esto//////////public string
EstadoDePuentes(string iniciador)

    public string climaentijuana()  ////para la aplicacion web hay que
quitar esto....
    {
        string iniciador = "tijuana";  //// para ejemplo le ponemos de una
vez el iniciador (en android)

        if (iniciador == "tijuana")
        {
            string a = "No Existe Conexión al servidor";

            // int i = 0;

            XmlTextReader reader = new
XmlTextReader("http://espanol.weather.com/rss-Tijuana-MXBC0005");
            XmlNodeType type;
            while (reader.Read())
            {
                type = reader.NodeType;

                if (type == XmlNodeType.Element)
                {
                    if (reader.Name == "item")
                    {
                        while (reader.Read())

                            if (reader.Name == "description")
                            {
                                reader.Read();

                                a = reader.Value;

```

```

        a = a.Replace("&deg;", "°");
        a = a.Replace("Para obtener más detalles...",
" ");
        reader.Close();

    }

}

}

}

// MessageBox.Show(i.ToString());
reader.Close();

return a;
}

return ("no existe conexion");
}

//////// se acaba el method

[WebMethod]

////////para la aplicacion android quitamos esto////////public string
EstadoDePuentes(string iniciador)

    public string alertaamericas()    ////para la aplicacion web hay que
quitar esto....
    {
        string iniciador = "americas";    //// para ejemplo le ponemos de una
vez el iniciador (en android)
        string diadelasemana = "";
        diadelasemana = dt.DayOfWeek.ToString();

        if (iniciador == "americas")
        {

            try
            {

                // Creando objetos de datos.
                System.Data.DataSet ds = new System.Data.DataSet();

```

```

        System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT tiempoactual FROM
estadisticasamericas WHERE dia = 'hoy' ", " Server=198.216.59.184;User Id
=prueba; password=****; Initial Catalog=clientes;");

        // Añadiendo el parámetro
        da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

        da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

        // Se recuperan los datos
        da.Fill(ds, "usuarios");

        // Se verifica la existencia del registro
        if (ds.Tables["usuarios"].Rows.Count != 0)
        {
            string tiempoactual = "";

            tiempoactual =
ds.Tables["usuarios"].Rows[0]["tiempoactual"].ToString();

            return tiempoactual;
        }
        else
        {
            string resultado3 = "ERROR";
            return resultado3;
        }
    }

    catch (System.Exception ex)
    {
        return ex.ToString();
    }

}

return ("no existe conexion");
}

```

```

//se acaba el webmethod

[WebMethod]

////////para la aplicacion android quitamos esto////////public string
EstadoDePuentes(string iniciador)

    public string alertacentro()    ////para la aplicacion web hay que quitar
esto....
    {
        string iniciador = "centro";    //// para ejemplo le ponemos de una vez
el iniciador (en android)
        string diadelasemana = "";
        diadelasemana = dt.DayOfWeek.ToString();

        if (iniciador == "centro")
        {

            try
            {

                // Creando objetos de datos.
                System.Data.DataSet ds = new System.Data.DataSet();
                System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT tiempoactual FROM estadisticascentro
WHERE dia = 'hoy' ", " Server=198.216.59.184;User Id =prueba; password=****;
Initial Catalog=clientes;");

                // Añadiendo el parámetro
                da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

                da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

                // Se recuperan los datos
                da.Fill(ds, "usuarios");

                // Se verifica la existencia del registro
                if (ds.Tables["usuarios"].Rows.Count != 0)
                {
                    string tiempoactual = "";

                    tiempoactual =
ds.Tables["usuarios"].Rows[0]["tiempoactual"].ToString();

                    return tiempoactual;
                }
            }
        }
    }
}

```

```

    }
    else
    {
        string resultado3 = "ERROR";
        return resultado3;
    }

}

catch (System.Exception ex)
{
    return ex.ToString();
}

}

return ("no existe conexion");
}

//se acaba el webmethod

[WebMethod]

/////////para la aplicacion android quitamos esto/////////public string
EstadoDePuentes(string iniciador)

public string alertaysleta()  ////para la aplicacion web hay que quitar
esto....
{
    string iniciador = "ysleta";  //// para ejemplo le ponemos de una vez
    el iniciador (en android)
    string diadelasemana = "";
    diadelasemana = dt.DayOfWeek.ToString();

    if (iniciador == "ysleta")
    {

        try
        {

            // Creando objetos de datos.
            System.Data.DataSet ds = new System.Data.DataSet();
            System.Data.SqlClient.SqlDataAdapter da = new
            System.Data.SqlClient.SqlDataAdapter("SELECT tiempoactual FROM estadisticasysleta
            WHERE dia = 'hoy' ", " Server=198.216.59.184;User Id =prueba; password=****;
            Initial Catalog=clientes;");

```

```

        // Añadiendo el parámetro
        da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

        da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

        // Se recuperan los datos
        da.Fill(ds, "usuarios");

        // Se verifica la existencia del registro
        if (ds.Tables["usuarios"].Rows.Count != 0)
        {
            string tiempoactual = "";

            tiempoactual =
ds.Tables["usuarios"].Rows[0]["tiempoactual"].ToString();

            return tiempoactual;
        }
        else
        {
            string resultado3 = "ERROR";
            return resultado3;
        }
    }

    catch (System.Exception ex)
    {
        return ex.ToString();
    }

    return ("no existe conexion");
}

//se acaba el webmethod
[WebMethod]

```



```

//////////para la aplicacion android quitamos esto//////////public string
EstadoDePuentes(string iniciador)

    public string alertaisidro()    ////para la aplicacion web hay que quitar
esto....
    {
        string iniciador = "isidro";    //// para ejemplo le ponemos de una vez
el iniciador (en android)
        string diadelasemana = "";
        diadelasemana = dt.DayOfWeek.ToString();

        if (iniciador == "isidro")
        {

            try
            {

                // Creando objetos de datos.
                System.Data.DataSet ds = new System.Data.DataSet();
                System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT tiempoactual FROM estadisticasisidro
WHERE dia = 'hoy' ", " Server=198.216.59.184;User Id =prueba; password=****;
Initial Catalog=clientes;");

                // Añadiendo el parámetro
                da.SelectCommand.Parameters.Add("@diadelasemana",
System.Data.SqlDbType.VarChar);

                da.SelectCommand.Parameters["@diadelasemana"].Value =
diadelasemana.ToString();

                // Se recuperan los datos
                da.Fill(ds, "usuarios");

                // Se verifica la existencia del registro
                if (ds.Tables["usuarios"].Rows.Count != 0)
                {
                    string tiempoactual = "";

                    tiempoactual =
ds.Tables["usuarios"].Rows[0]["tiempoactual"].ToString();

                    return tiempoactual;

                }
            }
            else
            {
                string resultado3 = "ERROR";
                return resultado3;

            }
        }
    }

```

```
    }  
    catch (System.Exception ex)  
    {  
        return ex.ToString();  
    }  
  
    }  
  
    return ("no existe conexion");  
}  
  
//se acaba el webmethod  
  
}  
  
}
```

Anexo 3.2

Web service para página web

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Xml;
using System.Data.SqlClient;

namespace MiServicioWeb2 ///////////////////////////////////////////////////WEB
BROWSER////////////////////////////////////
{
    /// <summary>
    /// Descripción breve de Service1
    /// </summary>
    [WebService(Namespace = "http://tempuri.org/",
Name="ServicioWeb",Description="By Juan")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    // Para permitir que se llame a este servicio Web desde un script, usando
    ASP.NET AJAX, quite la marca de comentario de la línea siguiente.
    // [System.Web.Script.Services.ScriptService]
    public class Service1 : System.Web.Services.WebService
    {

        [WebMethod]

        public string valor(string iniciador)
        {
            string resultado = "";

            try
            {

                // Creando objetos de datos.
                System.Data.DataSet ds = new System.Data.DataSet();
                System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT compra, venta FROM Usuarios WHERE
casadecambio = @casadecambio ", " Server=198.216.59.184; User Id =prueba;
password=****; Initial Catalog=clientes;");

                // Añadiendo el parámetro
da.SelectCommand.Parameters.Add("@casadecambio",
System.Data.SqlDbType.VarChar);

                da.SelectCommand.Parameters["@casadecambio"].Value = iniciador;


                // Se recuperan los datos
da.Fill(ds, "usuarios");

                // Se verifica la existencia del registro
if (ds.Tables["usuarios"].Rows.Count != 0)
{

```

```

        string cantidadcompra;
        string cantidadventa;
        string cantidadtotal;

        cantidadcompra =
ds.Tables["usuarios"].Rows[0]["compra"].ToString();
        cantidadventa =
ds.Tables["usuarios"].Rows[0]["venta"].ToString();

        cantidadtotal = string.Concat("Compra: $",cantidadcompra,"
Venta: $", cantidadventa);
        return cantidadtotal;

    }
    else
    {
        string resultado3 = "ERROR";
        return resultado3;

    }

}

catch (System.Exception ex)
{
    return ex.ToString();
}

}

[WebMethod]

public string guardarcambio(string compra, string venta, string casa)
{
    string resultado="ERROR";
    string resultado2="OK";

    try
    {

        System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184;User Id =prueba;
password=****; Initial Catalog=clientes;");
        System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

        // Estableciendo propiedades
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE usuarios SET compra = @compra , venta =
@venta WHERE casadecambio = @casadecambio ";
    }
}

```

```

        cmd.Parameters.Add("@casadecambio",
System.Data.SqlDbType.VarChar);
        cmd.Parameters.Add("@compra", System.Data.SqlDbType.VarChar);
        cmd.Parameters.Add("@venta", System.Data.SqlDbType.VarChar);

        cmd.Parameters["@casadecambio"].Value = casa;
        cmd.Parameters["@compra"].Value = compra;
        cmd.Parameters["@venta"].Value = venta;

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        return resultado2;
    }

    catch (System.Exception ex)
    {
        return resultado;
    }

}

[WebMethod]

public string casadecambio(string usuario, string password)
{
    string b="error";

    try
    {
        // Creando objetos de datos.
        System.Data.DataSet ds = new System.Data.DataSet();
        System.Data.SqlClient.SqlDataAdapter da = new
System.Data.SqlClient.SqlDataAdapter("SELECT * FROM Usuarios WHERE NickName =
@nickname AND Contraseña = @contrasena ", " Server=198.216.59.184;User Id
=prueba; password=****; Initial Catalog=clientes;");

        // Añadiendo el parámetro
        da.SelectCommand.Parameters.Add("@nickname",
System.Data.SqlDbType.VarChar);
        da.SelectCommand.Parameters.Add("@contrasena",
System.Data.SqlDbType.VarChar);
        da.SelectCommand.Parameters["@nickname"].Value = usuario;
        da.SelectCommand.Parameters["@contrasena"].Value = password;

        // Se recuperan los datos
        da.Fill(ds, "usuarios");
    }
}

```

```

        // Se verifica la existencia del registro
        if (ds.Tables["usuarios"].Rows.Count != 0)
        {
            string resultado="correcto";

            resultado=
ds.Tables["usuarios"].Rows[0]["casadecambio"].ToString();

            return resultado;

        }
        else
        {
            string resultado="ERROR";
            return resultado;

        }

    }

    catch (System.Exception ex)
    {

    }

    return b;
}

[WebMethod]
public string EstadoDePuentes(string iniciador)
{

    if (iniciador == "americas")
    {
        string a = "No Existe Conexión al Puente";

        ////////////juarez americas

        int i = 0;

        XmlTextReader reader = new
XmlTextReader("http://apps.cbp.gov/bwt/bwt.xml");

```

```

XmlNodeType type;
while (reader.Read())
{
    type = reader.NodeType;

    if (type == XmlNodeType.Element)
    {
        if (reader.Name == "port_number")
        {
            reader.Read();

            if (reader.Value == "240201")
            {
                reader.Read();

                while (reader.Read())

                    if (reader.Name == "passenger_vehicle_lanes")

                        {
                            while (reader.Read())

                                if (reader.Name == "delay_minutes")
                                {
                                    reader.Read();
                                    i++;
                                    if (i == 1)
                                    {

                                        a = reader.Value;
                                        reader.Close();

                                    }
                                }
                            }
                        }
                    }
                }
            }
        }

    }

}

}

}

// MessageBox.Show(i.ToString());
reader.Close();

return a;
}

```



```

if (iniciador == "centro")
{

    string a = "No Existe Conexión al Puente";

    //////////// juarez centro

    int i = 0;

    XmlTextReader reader = new
XmlTextReader("http://apps.cbp.gov/bwt/bwt.xml");
    XmlNodeType type;
    while (reader.Read())
    {
        type = reader.NodeType;

        if (type == XmlNodeType.Element)
        {
            if (reader.Name == "port_number")
            {

                reader.Read();

                if (reader.Value == "240202")
                {

                    reader.Read();

                    while (reader.Read())

                        if (reader.Name == "passenger_vehicle_lanes")

                            {

                                while (reader.Read())

                                    if (reader.Name == "delay_minutes")
                                    {
                                        reader.Read();
                                        i++;
                                        if (i == 1)
                                        {

                                            a = reader.Value;
                                            reader.Close();

                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }

}

```

```

        }

    }

    // MessageBox.Show(i.ToString());
    reader.Close();

    return a;
}

if (iniciador == "zaragoza")
{

    string a = "No Existe Conexión al Puente";

    ////////////juarez zaragoza

    int i = 0;

    XmlTextReader reader = new
    XmlTextReader("http://apps.cbp.gov/bwt/bwt.xml");
    XmlNodeType type;
    while (reader.Read())
    {
        type = reader.NodeType;

        if (type == XmlNodeType.Element)
        {
            if (reader.Name == "port_number")
            {

                reader.Read();

                if (reader.Value == "240203")
                {

                    reader.Read();

                    while (reader.Read())

                        if (reader.Name == "passenger_vehicle_lanes")

                            {

                                while (reader.Read())

                                    if (reader.Name == "delay_minutes")
                                    {
                                        reader.Read();
                                        i++;
                                        if (i == 1)
                                        {

                                            a = reader.Value;
                                            reader.Close();

```

```

        }

    }

}

}

}

    }

}

    reader.Close();

    return a;
}

if (iniciador == "san ysidro")
{
    string a = "No Existe Conexión al Puente";

    ////////////puente san ysidro tijuana

    int i = 0;

    XmlTextReader reader = new
    XmlTextReader("http://apps.cbp.gov/bwt/bwt.xml");
    XmlNodeType type;
    while (reader.Read())
    {
        type = reader.NodeType;

        if (type == XmlNodeType.Element)
        {
            if (reader.Name == "port_number")
            {
                reader.Read();

                if (reader.Value == "250401")
                {
                    reader.Read();

                    while (reader.Read())

                        if (reader.Name == "passenger_vehicle_lanes")

```

```

        {
            while (reader.Read())
            {
                if (reader.Name == "delay_minutes")
                {
                    reader.Read();
                    i++;
                    if (i == 1)
                    {
                        a = reader.Value;
                        reader.Close();
                    }
                }
            }
        }

    }

    reader.Close();

    return a;
}

//////////////////////////////////////// clima actual en juarez
if (iniciador == "juarez")
{
    string a = "No Existe Conexión al servidor";

    // int i = 0;

    XmlTextReader reader = new
    XmlTextReader("http://espanol.weather.com/rss-Ciudad-Juarez-MXCA0026");
    XmlNodeType type;
    while (reader.Read())
    {
        type = reader.NodeType;

        if (type == XmlNodeType.Element)
        {
            if (reader.Name == "item")
            {

```

```

        while (reader.Read())
        {
            if (reader.Name == "description")
            {
                reader.Read();

                a = reader.Value;

                a = a.Replace("&deg;", "°");
                a = a.Replace("Para obtener más
detalles...", " ");

                reader.Close();

            }

        }

    }

    reader.Close();

    return a;
}

////////// clima actual en tijuana

if (iniciador == "tijuana")
{
    string a = "No Existe Conexión al servidor";

    XmlTextReader reader = new
XmlTextReader("http://espanol.weather.com/rss-Tijuana-MXBC0005");
    XmlNodeType type;
    while (reader.Read())
    {
        type = reader.NodeType;

        if (type == XmlNodeType.Element)
        {
            if (reader.Name == "item")
            {
                while (reader.Read())

                    if (reader.Name == "description")
                    {

```

```

        reader.Read();

        a = reader.Value;

        a = a.Replace("&deg;", "°");
        a = a.Replace("Para obtener más detalles...",
" ");

        reader.Close();

    }

}

}

}

reader.Close();

return a;

}

return ("no existe conexion");

}

}

}

```

Anexo 3.3

Captura de tiempos para estadísticas

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Web;
using System.Xml;
using System.Data.SqlClient;
using System.IO;
using System.Drawing.Printing;
using System.Diagnostics;

namespace robot_tiempos
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            Timer timer = new Timer();
            timer.Tick += new EventHandler(timer_Tick);
            timer.Interval = (300000); // cada 5 minutos
            timer.Enabled = true;
            timer.Start();
        }

        void timer_Tick(object sender, EventArgs e)
        {
            button1_Click(null, null);
        }

        Process.Start("C:/Users/MELINITA/Documents/ScreenShotEvolucionPorHora.exe");
    }

    private void button1_Click(object sender, EventArgs e)
    {
        DateTime dt = DateTime.Now;

        int tiempototal = 0;
        int tiempohrs = 0;
        int tiempomins = 0;
        String tiempoletra = "";
        String temporal = "";
        char[] tiempochar;

        { ////////////Tijuana San Ysidro

            textBox1.Text = dt.TimeOfDay.ToString();

            string tiempo = "No Existe Conexión al Puente";

```



```

int i = 0;

XmlTextReader reader = new
XmlTextReader("http://apps.cbp.gov/bwt/bwt.xml");
XmlNodeType type;
try{
while (reader.Read())
{
    type = reader.NodeType;

    if (type == XmlNodeType.Element)
    {
        if (reader.Name == "port_number")
        {

            reader.Read();

            if (reader.Value == "250401")
            {

                reader.Read();

                while (reader.Read())

                    if (reader.Name == "passenger_vehicle_lanes")
                    {
                        while (reader.Read())

                            if (reader.Name == "delay_minutes")
                            {
                                reader.Read();
                                i++;
                                if (i == 1)
                                {

                                    tiempo = reader.Value;
                                    reader.Close();
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
}
}

```

```

        reader.Close();
    }
    catch (System.Exception ex) // del try de while reader
    {
        MessageBox.Show(ex.Message);
    }

    string horadelDia = "";
    string diadelaSemana = "";

    if (dt.Hour == 7 && dt.Minute <= 25)
    {
        horadelDia = "sieteam";
    }
    if (dt.Hour == 9 && dt.Minute <= 25)
    {
        horadelDia = "nueveam";
    }
    if (dt.Hour == 11 && dt.Minute <= 25)
    {
        horadelDia = "onceam";
    }
    if (dt.Hour == 13 && dt.Minute <= 25)
    {
        horadelDia = "unapm";
    }
    if (dt.Hour == 15 && dt.Minute <= 25)
    {
        horadelDia = "trespm";
    }
    if (dt.Hour == 17 && dt.Minute <= 25)
    {
        horadelDia = "cincopm";
    }
    if (dt.Hour == 19 && dt.Minute <= 25)
    {
        horadelDia = "sietepm";
    }
    if (dt.Hour == 21 && dt.Minute <= 25)
    {
        horadelDia = "nuevepm";
    }

    diadelaSemana = dt.DayOfWeek.ToString();

    if (horadelDia != "")
    {
        try
        {
            System.Data.SqlClient.SqlConnection conn = new
            System.Data.SqlClient.SqlConnection(" Server=198.216.59.184; User Id =prueba;
            password=****; Initial Catalog=clientes;");
            System.Data.SqlClient.SqlCommand cmd = new
            System.Data.SqlClient.SqlCommand();

            // Estableciendo propiedades
            cmd.Connection = conn;
            cmd.CommandText = "UPDATE estadisticasisidro SET " +
            horadelDia + " = @hora WHERE dia = '" + diadelaSemana + "' ";

```

```

        cmd.Parameters.Add("@hora", System.Data.SqlDbType.Text);

        //cmd.Parameters["@nueveam"].Value =
long.Parse(textBox1.Text);
        cmd.Parameters["@hora"].Value = tiempo.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        //////////////////////////////////

        // Estableciendo propiedades          // la hora de
captura...
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasisidro SET
horadecaptura = @horadecaptura WHERE dia = '" + diadelasemana + "' ";

        cmd.Parameters.Add("@horadecaptura",
System.Data.SqlDbType.Text);

        cmd.Parameters["@horadecaptura"].Value =
DateTime.Now.ToString("HH:mm:ss tt");

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

//////////////////////////////// a convertir el tiempo capturado en suma de minutos

textBox2.Text = tiempo.ToString();

// 0123456789
// 1 hrs
// 1 min
// 10 min
// 1 hrs 1 min
// 1 hrs 10 min
// 0123456789
tiempoleta = textBox2.Text;
temporal = textBox2.Text;

tiempochar = tiempoleta.ToCharArray();

```

```

try
{
    if (tiempochar[2].ToString().Equals("h"))
    {
        tiempohrs = Convert.ToInt32(temporal.Remove(2));
        tiempoletra = textBox2.Text;
        temporal = textBox2.Text;
    }
    if (tiempochar[2].ToString().Equals("m"))
    {
        tiempomins = Convert.ToInt32(temporal.Remove(2));
        tiempoletra = textBox2.Text;
        temporal = textBox2.Text;
    }
    if (tiempochar[3].ToString().Equals("m"))
    {
        tiempomins = Convert.ToInt32(temporal.Remove(3));
        tiempoletra = textBox2.Text;
        temporal = textBox2.Text;
    }

    // 0123456789
    // 1 hrs
    // 1 min
    // 10 min
    // 1 hrs 1 min
    // 1 hrs 10 min
    // 0123456789

    if (temporal.Length > 6)
    {
        if (tiempochar[8].ToString().Equals("m"))
        {
            tiempomins = Convert.ToInt32(tiempochar[6]) - 48;
            tiempoletra = textBox2.Text;
            temporal = textBox2.Text;
        }
        if (tiempochar[9].ToString().Equals("m"))
        {
            tiempomins = (Convert.ToInt32(tiempochar[6]) - 48) * 10 +
            (Convert.ToInt32(tiempochar[7] - 48));
            tiempoletra = textBox2.Text;
            temporal = textBox2.Text;
        }
    }
} // del try
catch (System.Exception ex)
{
    MessageBox.Show(ex.Message);
}

tiempototal = tiempohrs * 60 + tiempomins;
tiempohrs = 0;
tiempomins = 0;
//////////introducir el tiempo en minutos a la hora

```

actual

```

try
{
    System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184; User Id =prueba;
password=****; Initial Catalog=clientes;");
    System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

    // Estableciendo propiedades        // el tiempo actual en
minutos...
    cmd.Connection = conn;
    cmd.CommandText = "UPDATE estadisticasisidro SET tiempoactual =
@tiempototal WHERE dia = 'hoy' ";

    cmd.Parameters.Add("@tiempototal", System.Data.SqlDbType.Text);

    cmd.Parameters["@tiempototal"].Value = tiempototal.ToString();

    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    // Estableciendo propiedades        // el tiempo actual completo
con horas y minutos...
    cmd.Connection = conn;
    cmd.CommandText = "UPDATE estadisticasisidro SET
tiempoactualcompleto = @tiempototalcompleto WHERE dia = 'hoy'";

    cmd.Parameters.Add("@tiempototalcompleto",
System.Data.SqlDbType.Text);

    cmd.Parameters["@tiempototalcompleto"].Value = tiempo.ToString();

    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    // Estableciendo propiedades        // la hora de captura...
    cmd.Connection = conn;
    cmd.CommandText = "UPDATE estadisticasisidro SET capturahoy =
@horadecaptura WHERE dia= '' + diadelasemana + '' ";

    cmd.Parameters.Add("@horadecaptura", System.Data.SqlDbType.Text);

    cmd.Parameters["@horadecaptura"].Value =
DateTime.Now.ToString("HH:mm:ss tt");

    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    // Estableciendo propiedades        // el tiempo actual completo
en dia de la semana
    cmd.Connection = conn;
    cmd.CommandText = "UPDATE estadisticasisidro SET
tiempoactualcompleto = @tiempoactualcompleto WHERE dia = '' + diadelasemana + ''
";

```

```

        cmd.Parameters.Add("@tiempoactualcompleto",
System.Data.SqlDbType.Text);

        cmd.Parameters["@tiempoactualcompleto"].Value =
tiempo.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

```

```

    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

```

```

} //se acaba San Ysidro

```

```

{ ////////////juarez americas

    textBox1.Text = dt.TimeOfDay.ToString();

    string tiempo = "No Existe Conexión al Puente";

```

```

        int i = 0;

        XmlTextReader reader = new
XmlTextReader("http://apps.cbp.gov/bwt/bwt.xml");
        XmlNodeType type;
        try{

            while (reader.Read())
            {
                type = reader.NodeType;

```

```

if (type == XmlNodeType.Element)
{
    if (reader.Name == "port_number")
    {
        reader.Read();

        if (reader.Value == "240201")
        {
            reader.Read();

            while (reader.Read())
            {
                if (reader.Name == "passenger_vehicle_lanes")
                {
                    while (reader.Read())
                    {
                        if (reader.Name == "delay_minutes")
                        {
                            reader.Read();
                            i++;
                            if (i == 1)
                            {
                                tiempo = reader.Value;
                                reader.Close();
                            }
                        }
                    }
                }
            }
        }
    }
}

}

}

reader.Close();
}
catch (System.Exception ex) // del try de while reader
{
    MessageBox.Show(ex.Message);
}

string horadelDia = "";
string diadelaSemana = "";

if (dt.Hour == 7 && dt.Minute <= 25)
{

```

```

        horadeldia = "sieteam";
    }

    if (dt.Hour == 9 && dt.Minute <= 25)
    {
        horadeldia = "nueveam";
    } if (dt.Hour == 11 && dt.Minute <= 25)
    {
        horadeldia = "onceam";
    }
    if (dt.Hour == 13 && dt.Minute <= 25)
    {
        horadeldia = "unapm";
    } if (dt.Hour == 15 && dt.Minute <= 25)
    {
        horadeldia = "trespm";
    } if (dt.Hour == 17 && dt.Minute <= 25)
    {
        horadeldia = "cincopm";
    } if (dt.Hour == 19 && dt.Minute <= 25)
    {
        horadeldia = "sietepm";
    }
    if (dt.Hour == 21 && dt.Minute <= 25)
    {
        horadeldia = "nuevepm";
    }

    diadelasemana = dt.DayOfWeek.ToString();

    if (horadeldia != "")
    {
        try
        {
            System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184; User Id =prueba;
password=****; Initial Catalog=clientes;");
            System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

            // Estableciendo propiedades
            cmd.Connection = conn;
            cmd.CommandText = "UPDATE estadisticasamericas SET " +
horadeldia + " = @hora WHERE dia = '" + diadelasemana + "' ";

            cmd.Parameters.Add("@hora", System.Data.SqlDbType.Text);

            cmd.Parameters["@hora"].Value = tiempo.ToString();

            conn.Open();
            cmd.ExecuteNonQuery();
            conn.Close();

            //////////////////////////////////

            // Estableciendo propiedades          // el la hora de
captura...
            cmd.Connection = conn;

```



```

        cmd.CommandText = "UPDATE estadisticasamericas SET
horadecaptura = @horadecaptura WHERE dia = '" + diadelasemana + "' ";

        cmd.Parameters.Add("@horadecaptura",
System.Data.SqlDbType.Text);

        cmd.Parameters["@horadecaptura"].Value =
DateTime.Now.ToString("HH:mm:ss tt");

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

////////// a convertir el tiempo capturado en suma de minutos

textBox2.Text = tiempo.ToString();

//0123456789
// 1 hrs
// 1 min
// 10 min
// 1 hrs 1 min
// 1 hrs 10 min
// 0123456789
tiempoletra = textBox2.Text;
temporal = textBox2.Text;

tiempochar = tiempoletra.ToCharArray();

try {
if (tiempochar[2].ToString().Equals("h"))
{
    tiempohrs = Convert.ToInt32(temporal.Remove(2));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}
if (tiempochar[2].ToString().Equals("m"))
{
    tiempomins = Convert.ToInt32(temporal.Remove(2));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}
if (tiempochar[3].ToString().Equals("m"))
{
    tiempomins = Convert.ToInt32(temporal.Remove(3));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}
}

```

```

    }

    // 0123456789
    // 1 hrs
    // 1 min
    // 10 min
    // 1 hrs 1 min
    // 1 hrs 10 min
    // 0123456789

    if (temporal.Length > 6)
    {
        if (tiempochar[8].ToString().Equals("m"))
        {
            tiempomins = Convert.ToInt32(tiempochar[6]) - 48;
            tiempoletra = textBox2.Text;
            temporal = textBox2.Text;

        }
        if (tiempochar[9].ToString().Equals("m"))
        {
            tiempomins = (Convert.ToInt32(tiempochar[6]) - 48) * 10 +
(Convert.ToInt32(tiempochar[7] - 48));
            tiempoletra = textBox2.Text;
            temporal = textBox2.Text;
        }
    }
    } // del try
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

    tiempototal = tiempohrs * 60 + tiempomins;
    tiempohrs = 0;
    tiempomins = 0;
    ///////////////////////////////////////////////////introducir el tiempo en minutos a la
hora actual

    try
    {
        System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184; User Id =prueba;
password=****; Initial Catalog=clientes;");
        System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

        // Estableciendo propiedades        // el tiempo actual en
minutos...

        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasamericas SET
tiempoactual = @tiempototal WHERE dia = 'hoy' ";

        cmd.Parameters.Add("@tiempototal",
System.Data.SqlDbType.Text);

```

```

        //cmd.Parameters["@nueveam"].Value =
long.Parse(textBox1.Text);
        cmd.Parameters["@tiempototal"].Value = tiempototal.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        // Estableciendo propiedades        // el tiempo actual
completo con horas y minutos...
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasamericas SET
tiempoactualcompleto = @tiempototalcompleto WHERE dia = 'hoy' ";

        cmd.Parameters.Add("@tiempototalcompleto",
System.Data.SqlDbType.Text);

        //cmd.Parameters["@nueveam"].Value =
long.Parse(textBox1.Text);
        cmd.Parameters["@tiempototalcompleto"].Value =
tiempo.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        // Estableciendo propiedades        // el la hora de
captura...
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasamericas SET capturahoy
= @horadecaptura WHERE dia= '' + diadelasemana + '' ";

        cmd.Parameters.Add("@horadecaptura",
System.Data.SqlDbType.Text);

        cmd.Parameters["@horadecaptura"].Value =
DateTime.Now.ToString("HH:mm:ss tt");

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        // Estableciendo propiedades        // el tiempo actual
completo en dia de la semana
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasamericas SET
tiempoactualcompleto = @tiempoactualcompleto WHERE dia = '' + diadelasemana + ''
";

        cmd.Parameters.Add("@tiempoactualcompleto",
System.Data.SqlDbType.Text);

        cmd.Parameters["@tiempoactualcompleto"].Value =
tiempo.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

```

```

    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

} //se acaba americas

{ ////////////juarez centro

    textBox1.Text = dt.TimeOfDay.ToString(); ;

    string tiempo = "No Existe Conexión al Puente";

    int i = 0;

    XmlTextReader reader = new
    XmlTextReader("http://apps.cbp.gov/bwt/bwt.xml");
    XmlNodeType type;

    try
    {
        while (reader.Read())
        {
            type = reader.NodeType;

            if (type == XmlNodeType.Element)
            {
                if (reader.Name == "port_number")
                {
                    reader.Read();

                    if (reader.Value == "240202")
                    {
                        reader.Read();

                        while (reader.Read())

```

```

        if (reader.Name ==
"passenger_vehicle_lanes")
        {
            while (reader.Read())

                if (reader.Name ==
"delay_minutes")
                {
                    reader.Read();
                    i++;
                    if (i == 1)
                    {
                        tiempo = reader.Value;
                        reader.Close();
                    }
                }
            }
        }
    }

}

}

}

reader.Close();
}
catch (System.Exception ex) // del try de while reader
{
    MessageBox.Show(ex.Message);
}

string horadelDia = "";
string diadelaSemana = "";

if (dt.Hour == 7 && dt.Minute <= 25)
{
    horadelDia = "sieteam";
}
if (dt.Hour == 9 && dt.Minute <= 25)
{
    horadelDia = "nueveam";
}
if (dt.Hour == 11 && dt.Minute <= 25)
{
    horadelDia = "onceam";
}
if (dt.Hour == 13 && dt.Minute <= 25)
{
    horadelDia = "unapm";
}
if (dt.Hour == 15 && dt.Minute <= 25)
{
    horadelDia = "trespm";
}

```

```

    } if (dt.Hour == 17 && dt.Minute <= 25)
    {
        horadeldia = "cincopm";
    } if (dt.Hour == 19 && dt.Minute <= 25)
    {
        horadeldia = "sietepm";
    }
    if (dt.Hour == 21 && dt.Minute <= 25)
    {
        horadeldia = "nuevepm";
    }

    diadelasemana = dt.DayOfWeek.ToString();

    if (horadeldia != "")
    {
        try
        {
            System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184; User Id =prueba;
password=****; Initial Catalog=clientes;");
            System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

            // Estableciendo propiedades
            cmd.Connection = conn;
            cmd.CommandText = "UPDATE estadisticascentro SET " +
horadeldia + " = @hora WHERE dia = '" + diadelasemana + "' ";

            cmd.Parameters.Add("@hora", System.Data.SqlDbType.Text);

            cmd.Parameters["@hora"].Value = tiempo.ToString();

            conn.Open();
            cmd.ExecuteNonQuery();
            conn.Close();

            ///////////////////////////////////

            // Estableciendo propiedades        // el la hora de
captura...
            cmd.Connection = conn;
            cmd.CommandText = "UPDATE estadisticascentro SET
horadecaptura = @horadecaptura WHERE dia = '" + diadelasemana + "' ";

            cmd.Parameters.Add("@horadecaptura",
System.Data.SqlDbType.Text);

            cmd.Parameters["@horadecaptura"].Value =
DateTime.Now.ToString("HH:mm:ss tt");

            conn.Open();
            cmd.ExecuteNonQuery();
            conn.Close();

```

```

    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

////////// a convertir el tiempo capturado en suma de minutos

textBox2.Text = tiempo.ToString();

//0123456789
// 1 hrs
// 1 min
// 10 min
// 1 hrs 1 min
// 1 hrs 10 min
// 0123456789
tiempoletra = textBox2.Text;
temporal = textBox2.Text;

tiempochar = tiempoletra.ToCharArray();

try {
if (tiempochar[2].ToString().Equals("h"))
{
    tiempohrs = Convert.ToInt32(temporal.Remove(2));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}
if (tiempochar[2].ToString().Equals("m"))
{
    tiempomins = Convert.ToInt32(temporal.Remove(2));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}
if (tiempochar[3].ToString().Equals("m"))
{
    tiempomins = Convert.ToInt32(temporal.Remove(3));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}

// 0123456789
// 1 hrs
// 1 min
// 10 min
// 1 hrs 1 min
// 1 hrs 10 min
// 0123456789

if (temporal.Length > 6)
{
    if (tiempochar[8].ToString().Equals("m"))

```

```

{
    tiempomins = Convert.ToInt32(tiempochar[6]) - 48;
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;

}
if (tiempochar[9].ToString().Equals("m"))
{
    tiempomins = (Convert.ToInt32(tiempochar[6]) - 48) * 10 +
(Convert.ToInt32(tiempochar[7] - 48));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}
}
} // del try
catch (System.Exception ex)
{
    MessageBox.Show(ex.Message);
}

tiempototal = tiempohrs * 60 + tiempomins;
tiempohrs = 0;
tiempomins = 0;
//////////introducir el tiempo en minutos a la
hora actual

try
{
    System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184; User Id =prueba;
password=****; Initial Catalog=clientes;");
    System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

    // Estableciendo propiedades        // el tiempo actual en
minutos...
    cmd.Connection = conn;
    cmd.CommandText = "UPDATE estadisticascentro SET tiempoactual
= @tiempototal WHERE dia = 'hoy' ";

    cmd.Parameters.Add("@tiempototal",
System.Data.SqlDbType.Text);

    //cmd.Parameters["@nueveam"].Value =
long.Parse(textBox1.Text);
    cmd.Parameters["@tiempototal"].Value = tiempototal.ToString();

    conn.Open();
    cmd.ExecuteNonQuery();
    conn.Close();

    // Estableciendo propiedades        // el tiempo actual
completo con horas y minutos...
    cmd.Connection = conn;
    cmd.CommandText = "UPDATE estadisticascentro SET
tiempoactualcompleto = @tiempototalcompleto WHERE dia = 'hoy' ";

```



```

        cmd.Parameters.Add("@tiempototalcompleto",
System.Data.SqlDbType.Text);
        cmd.Parameters["@tiempototalcompleto"].Value =
tiempo.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

// Estableciendo propiedades        // el la hora de
captura...
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticascentro SET capturahoy =
@horadecaptura WHERE dia= '" + diadelasemana + "' ";

        cmd.Parameters.Add("@horadecaptura",
System.Data.SqlDbType.Text);

        cmd.Parameters["@horadecaptura"].Value =
DateTime.Now.ToString("HH:mm:ss tt");

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

// Estableciendo propiedades        // el tiempo actual
completo en dia de la semana
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticascentro SET
tiempoactualcompleto = @tiempoactualcompleto WHERE dia = '" + diadelasemana + "'
";

        cmd.Parameters.Add("@tiempoactualcompleto",
System.Data.SqlDbType.Text);

        cmd.Parameters["@tiempoactualcompleto"].Value =
tiempo.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

    } // se acaba centro

    { ////////////juarez zaragoza

    textBox1.Text = dt.TimeOfDay.ToString(); ;

    string tiempo = "No Existe Conexión al Puente";

    int i = 0;

    XmlTextReader reader = new
    XmlTextReader("http://apps.cbp.gov/bwt/bwt.xml");
    XmlNodeType type;
    try
    {
        while (reader.Read())
        {
            type = reader.NodeType;

            if (type == XmlNodeType.Element)
            {
                if (reader.Name == "port_number")
                {

                    reader.Read();

                    if (reader.Value == "240203")
                    {

                        reader.Read();

                        while (reader.Read())

                            if (reader.Name ==

"passenger_vehicle_lanes")

                                {
                                    while (reader.Read())

                                        if (reader.Name ==

"delay_minutes")

                                            {
                                                reader.Read();
                                                i++;
                                                if (i == 1)
                                                {

                                                    tiempo = reader.Value;
                                                    reader.Close();
                                                }
                                            }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
}

}

}

}

}

    reader.Close();
}
catch (System.Exception ex) // del try de while reader
{
    MessageBox.Show(ex.Message);
}

string horadeldia = "";
string diadelasemana = "";

if (dt.Hour == 7 && dt.Minute <= 25)
{
    horadeldia = "sieteam";
}
if (dt.Hour == 9 && dt.Minute <= 25)
{
    horadeldia = "nueveam";
}
if (dt.Hour == 11 && dt.Minute <= 25)
{
    horadeldia = "onceam";
}
if (dt.Hour == 13 && dt.Minute <= 25)
{
    horadeldia = "unapm";
}
if (dt.Hour == 15 && dt.Minute <= 25)
{
    horadeldia = "trespm";
}
if (dt.Hour == 17 && dt.Minute <= 25)
{
    horadeldia = "cincopm";
}
if (dt.Hour == 19 && dt.Minute <= 25)
{
    horadeldia = "sietepm";
}
if (dt.Hour == 21 && dt.Minute <= 25)
{
    horadeldia = "nuevepm";
}

diadelasemana = dt.DayOfWeek.ToString();

if (horadeldia != "")

```

```

{
    try
    {
        System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184; User Id =prueba;
password=****; Initial Catalog=clientes;");
        System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

        // Estableciendo propiedades
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasysleta SET " +
horadeldia + " = @hora WHERE dia = '" + diadelasemana + "' ";

        cmd.Parameters.Add("@hora", System.Data.SqlDbType.Text);

        cmd.Parameters["@hora"].Value = tiempo.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        ///////////

        // Estableciendo propiedades        // el la hora de
captura...
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasysleta SET
horadecaptura = @horadecaptura WHERE dia = '" + diadelasemana + "' ";

        cmd.Parameters.Add("@horadecaptura",
System.Data.SqlDbType.Text);

        cmd.Parameters["@horadecaptura"].Value =
DateTime.Now.ToString("HH:mm:ss tt");

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

////////// a convertir el tiempo capturado en suma de minutos

```

textBox2.Text = tiempo.ToString();

//0123456789
// 1 hrs
// 1 min
// 10 min
// 1 hrs 1 min
// 1 hrs 10 min
// 0123456789
tiempoletra = textBox2.Text;
temporal = textBox2.Text;

tiempochar = tiempoletra.ToCharArray();

try {
if (tiempochar[2].ToString().Equals("h"))
{
    tiempohrs = Convert.ToInt32(temporal.Remove(2));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}
if (tiempochar[2].ToString().Equals("m"))
{
    tiempomins = Convert.ToInt32(temporal.Remove(2));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}
if (tiempochar[3].ToString().Equals("m"))
{
    tiempomins = Convert.ToInt32(temporal.Remove(3));
    tiempoletra = textBox2.Text;
    temporal = textBox2.Text;
}

// 0123456789
// 1 hrs
// 1 min
// 10 min
// 1 hrs 1 min
// 1 hrs 10 min
// 0123456789

if (temporal.Length > 6)
{
    if (tiempochar[8].ToString().Equals("m"))
    {
        tiempomins = Convert.ToInt32(tiempochar[6]) - 48;
        tiempoletra = textBox2.Text;
        temporal = textBox2.Text;

    }
    if (tiempochar[9].ToString().Equals("m"))
    {
        tiempomins = (Convert.ToInt32(tiempochar[6]) - 48) * 10 +
        (Convert.ToInt32(tiempochar[7] - 48));
        tiempoletra = textBox2.Text;
        temporal = textBox2.Text;
    }
}

```

```

    }
    } // del try
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

    tiempototal = tiempohrs * 60 + tiempomins;
    tiempohrs = 0;
    tiempomins = 0;
    //////////////////////////////////////////////////introducir el tiempo en minutos a la
hora actual

    try
    {
        System.Data.SqlClient.SqlConnection conn = new
System.Data.SqlClient.SqlConnection(" Server=198.216.59.184; User Id =prueba;
password=****; Initial Catalog=clientes;");
        System.Data.SqlClient.SqlCommand cmd = new
System.Data.SqlClient.SqlCommand();

        // Estableciendo propiedades        // el tiempo actual en
minutos...
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasysleta SET tiempoactual
= @tiempototal WHERE dia = 'hoy' ";

        cmd.Parameters.Add("@tiempototal",
System.Data.SqlDbType.Text);

        //cmd.Parameters["@nueveam"].Value =
long.Parse(textBox1.Text);
        cmd.Parameters["@tiempototal"].Value = tiempototal.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        // Estableciendo propiedades        // el tiempo actual
completo con horas y minutos...
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasysleta SET
tiempoactualcompleto = @tiempototalcompleto WHERE dia = 'hoy' ";

        cmd.Parameters.Add("@tiempototalcompleto",
System.Data.SqlDbType.Text);

        cmd.Parameters["@tiempototalcompleto"].Value =
tiempo.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        // Estableciendo propiedades        // el la hora de
captura...
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasysleta SET capturahoy =
@horadecaptura WHERE dia= '" + diadelasemana + "' ";

```

```

        cmd.Parameters.Add("@horadecaptura",
System.Data.SqlDbType.Text);

        //cmd.Parameters["@nueveam"].Value =
long.Parse(textBox1.Text);
        cmd.Parameters["@horadecaptura"].Value =
DateTime.Now.ToString("HH:mm:ss tt");

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

        // Estableciendo propiedades        // el tiempo actual
completo en dia de la semana
        cmd.Connection = conn;
        cmd.CommandText = "UPDATE estadisticasysleta SET
tiempoactualcompleto = @tiempoactualcompleto WHERE dia = '" + diadelasemana + '"
";

        cmd.Parameters.Add("@tiempoactualcompleto",
System.Data.SqlDbType.Text);

        cmd.Parameters["@tiempoactualcompleto"].Value =
tiempo.ToString();

        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();

    }
    catch (System.Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

    } // se acaba zaragoza

}

// se acaba el click
private void Form1_Load(object sender, System.EventArgs e)
{
}

```

```
private void textBox3_TextChanged(object sender, EventArgs e)
{
}

private void textBox4_TextChanged(object sender, EventArgs e)
{
}

}
```


Anexo 3.4

Código de aplicación móvil

```

/// Pantalla Tiempos de Cruce
package com.example.webservice;

import android.os.Bundle;
import android.os.StrictMode;
import android.os.Vibrator;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.NotificationManager;

import android.content.Context;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.text.method.LinkMovementMethod;
import android.view.Menu;
import android.view.View;

import org.ksoap2.serialization.SoapObject;
import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapPrimitive;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

import android.widget.TextView;
import android.widget.Toast;

import android.graphics.PorterDuff;

public class GetServicio extends Activity implements View.OnClickListener {

    private static final String accionSoap =
"http://tempuri.org/EstadoDePuentes";
    private static final String accionSoap2 =
"http://tempuri.org/EstadoDePuentes2";
    private static final String accionSoap3 =
"http://tempuri.org/EstadoDePuentes3";
    private static final String accionSoap4 =
"http://tempuri.org/EstadoDePuentes4";
    private static final String accionSoap5 =
"http://tempuri.org/climaenjuarez";
    private static final String accionSoap6 =
"http://tempuri.org/climaentijuana";
    private static final String accionSoap7 = "http://tempuri.org/valor";

    private static final String Metodo = "EstadoDePuentes";
    private static final String Metodo2 = "EstadoDePuentes2";
    private static final String Metodo3= "EstadoDePuentes3";
    private static final String Metodo4= "EstadoDePuentes4";
    private static final String Metodo5= "climaenjuarez";
    private static final String Metodo6= "climaentijuana";
    private static final String Metodo7= "valor";

    private static final String namespace = "http://tempuri.org/";

```

```

        private static final String url =
"http://198.216.59.184/Servicio%20Android/service1.asmx";

        public String origen = "";
        public int origenint = 0;
        public int a = 0;  /// para el click por primera vez del boton de
clima

        TextView tv;
        NotificationManager nmx;
        static final int uniqueID=123123;

        @SuppressWarnings({ "NewApi", "NewApi", "NewApi" })
        @Override
        public void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);

            setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_NOSENSOR);

            setContentView(R.layout.main);
            View boton = findViewById(R.id.button1);
            boton.setOnClickListener(this);
            View boton2 = findViewById(R.id.button2);
            boton2.setOnClickListener(this);
            View boton3 = findViewById(R.id.button3);
            boton3.setOnClickListener(this);
            View boton4 = findViewById(R.id.button4);
            boton4.setOnClickListener(this);
            View boton5 = findViewById(R.id.button5);
            boton5.setOnClickListener(this);

            boton3.setVisibility(View.INVISIBLE);
            boton4.setVisibility(View.INVISIBLE);
            boton5.setVisibility(View.INVISIBLE);

            boton.getBackground().setColorFilter(0xFFB0C4DE,
PorterDuff.Mode.MULTIPLY);
            boton2.getBackground().setColorFilter(0xFFB0C4DE,
PorterDuff.Mode.MULTIPLY);
            boton3.getBackground().setColorFilter(0xFF00CED1,
PorterDuff.Mode.MULTIPLY);
            boton4.getBackground().setColorFilter(0xFF00CED1,
PorterDuff.Mode.MULTIPLY);
            boton5.getBackground().setColorFilter(0xFF00CED1,
PorterDuff.Mode.MULTIPLY);

            /// desactiva el modo estricto que impide hacer llamadas desde
la actividad main, es lo que impedía que funcionara en
/// el celular, solo funcionaba en el emulador.

            StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);

```

```
///// esto es necesario para que se vuelvan links las
ubicaciones, si no el usuario no las puede acceder
```

```
TextView t2 = (TextView) findViewById(R.id.TextView06);
t2.setMovementMethod(LinkMovementMethod.getInstance());
TextView t3 = (TextView) findViewById(R.id.TextView07);
t3.setMovementMethod(LinkMovementMethod.getInstance());
TextView t4 = (TextView) findViewById(R.id.TextView08);
t4.setMovementMethod(LinkMovementMethod.getInstance());
```

```
t2.setVisibility(View.INVISIBLE);
t3.setVisibility(View.INVISIBLE);
t4.setVisibility(View.INVISIBLE);
```

```
TextView texto11 = (TextView)findViewById(R.id.textView1);
TextView texto21 = (TextView)findViewById(R.id.textView2);
TextView texto31 = (TextView)findViewById(R.id.textView3);
TextView texto41 = (TextView)findViewById(R.id.textView4);
TextView texto51 = (TextView)findViewById(R.id.textView5);
TextView texto61 = (TextView)findViewById(R.id.textView6);
TextView texto71 = (TextView)findViewById(R.id.textView7);
TextView texto81 = (TextView)findViewById(R.id.textView8);
TextView texto91 =
(TextView)findViewById(R.id.casapatmos);
TextView texto101 =
(TextView)findViewById(R.id.TextView01);
TextView texto102 =
(TextView)findViewById(R.id.TextView02);
TextView texto103 =
(TextView)findViewById(R.id.TextView03);
TextView texto104 =
(TextView)findViewById(R.id.TextView04);
TextView texto105 =
(TextView)findViewById(R.id.TextView05);
```

```
texto11.setText("");
texto21.setText("");
texto31.setText("");
texto41.setText("");
texto51.setText("");
texto61.setText("");
texto71.setText("");
texto81.setText("");
texto91.setText("");
texto101.setText("");
texto102.setText("");
texto103.setText("");
texto104.setText("");
texto105.setText("");
```

```

    }

    public void BotonYes (View v){

        // boton para pruebas

    }////////// se acaba el boton Yes!

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);

        return true;
    }

    @Override
    public void onClick(View v) {

        View boton3 = findViewById(R.id.button3);
        View boton4 = findViewById(R.id.button4);
        View boton5 = findViewById(R.id.button5);

        boton3.setVisibility(View.VISIBLE);
        boton4.setVisibility(View.VISIBLE);
        boton5.setVisibility(View.VISIBLE);

        // TODO Auto-generated method stub
        TextView texto1 = (TextView)findViewById(R.id.textView1);
        TextView texto2 = (TextView)findViewById(R.id.textView2);
        TextView texto3 = (TextView)findViewById(R.id.textView3);
        TextView texto4 = (TextView)findViewById(R.id.textView4);
        TextView texto5 = (TextView)findViewById(R.id.textView5);
        TextView texto6 = (TextView)findViewById(R.id.textView6);
        TextView texto7 = (TextView)findViewById(R.id.textView7);
        TextView texto8 = (TextView)findViewById(R.id.casapatmos);
        TextView texto9 = (TextView)findViewById(R.id.TextView01);
        TextView texto10 =
(TextView)findViewById(R.id.TextView02);
        TextView texto11 =
(TextView)findViewById(R.id.TextView03);
        TextView texto12 =
(TextView)findViewById(R.id.TextView04);
        TextView texto13 =
(TextView)findViewById(R.id.TextView05);

        TextView t2 = (TextView) findViewById(R.id.TextView06);
        t2.setMovementMethod(LinkMovementMethod.getInstance());
        TextView t3 = (TextView) findViewById(R.id.TextView07);

```

```

t3.setMovementMethod(LinkMovementMethod.getInstance());
TextView t4 = (TextView) findViewById(R.id.TextView08);
t4.setMovementMethod(LinkMovementMethod.getInstance());
    t2.setVisibility(View.VISIBLE);
    t3.setVisibility(View.VISIBLE);
    t4.setVisibility(View.VISIBLE);

    if (v.getId() == findViewById(R.id.button1).getId()) {

        a=1;

        origen = "juarez";
        origenint =1;

        t2.setVisibility(View.VISIBLE);
        t3.setVisibility(View.VISIBLE); //son la palabra ubicacion
hay que volverlas visibles cuando halla casas de cambi en tijuana
        t4.setVisibility(View.VISIBLE);

        Vibrator vibrator =(Vibrator)
getSystemService(Context.VIBRATOR_SERVICE);

        vibrator.vibrate(100);

        /////Casas de Cambio
        try {

            // request
            SoapObject request7 = new SoapObject(namespace,
Metodo7);
            SoapObject request8 = new SoapObject(namespace,
Metodo7);
            SoapObject request9 = new SoapObject(namespace,
Metodo7);

            ///// //enviar parametros al metodo
            request7.addProperty("iniciador", "patmos");
            request8.addProperty("iniciador", "valor");
            request9.addProperty("iniciador", "mr. dolar");

```

```

        // modelo del sobre
        SoapSerializationEnvelope sobre7 = new
SoapSerializationEnvelope(
            SoapEnvelope.VER11);
        sobre7.dotNet = true;
        sobre7.setOutputSoapObject(request7);

        SoapSerializationEnvelope sobre8 = new
SoapSerializationEnvelope(
            SoapEnvelope.VER11);
        sobre8.dotNet = true;
        sobre8.setOutputSoapObject(request8);

        SoapSerializationEnvelope sobre9 = new
SoapSerializationEnvelope(
            SoapEnvelope.VER11);
        sobre9.dotNet = true;
        sobre9.setOutputSoapObject(request9);

        // modelo de transporte
        HttpTransportSE transporte7 = new
HttpTransportSE(url);
        HttpTransportSE transporte8 = new
HttpTransportSE(url);
        HttpTransportSE transporte9 = new

        // llamada
        transporte7.call(accionSoap7, sobre7);
        transporte8.call(accionSoap7, sobre8);
        transporte9.call(accionSoap7, sobre9);

        // Resultado
        sobre7.getResponse();
        sobre8.getResponse();
        sobre9.getResponse();

        SoapPrimitive resultado7 = (SoapPrimitive)
        SoapPrimitive resultado8 = (SoapPrimitive)
        SoapPrimitive resultado9 = (SoapPrimitive)

        texto8.setText("" + resultado7.toString());
        texto9.setText("Patmos:");

        texto10.setText("" + resultado8.toString());
        texto11.setText("Valor:");

        texto13.setText("" + resultado9.toString());
        texto12.setText("Mr. Dollar:");

```

```

    }

    catch (Exception e) {
        tv.setText(e.getMessage());
    }

}

/////el clima en juarez
try {

    // request
    SoapObject request5 = new

SoapObject(namespace, Metodo5);

    //enviar parametros al metodo

    // modelo del sobre
    SoapSerializationEnvelope sobre5 = new

SoapSerializationEnvelope(

        SoapEnvelope.VER11);
    sobre5.dotNet = true;
    sobre5.setOutputSoapObject(request5);

    // modelo de transporte
    HttpTransportSE transporte5 = new

HttpTransportSE(url);

    // llamada
    transporte5.call(accionSoap5, sobre5);

    // Resultado
    SoapPrimitive resultado5 =

(SoapPrimitive) sobre5.getResponse();

    texto7.setText("" +

resultado5.toString());

}

catch (Exception e) {
    tv.setText(e.getMessage());
}

}

/////el puente americas
try {

```



```

        // request
        SoapObject request = new SoapObject(namespace,
Metodo);

        // modelo del sobre
        SoapSerializationEnvelope sobre = new
SoapSerializationEnvelope(
            SoapEnvelope.VER11);
        sobre.dotNet = true;
        sobre.setOutputSoapObject(request);

        // modelo de transporte
        HttpTransportSE transporte = new

HttpTransportSE(url);

        // llamada
        transporte.call(accionSoap, sobre);

        // Resultado
        SoapPrimitive resultado1 = (SoapPrimitive)
sobre.getResponse();

        texto1.setText("Américas:");
        texto2.setText("" + resultado1.toString());

    }

    catch (Exception e) {
        tv.setText(e.getMessage());
    }

    /////////////// el puente del centro
    try {

        // request
        SoapObject request2 = new SoapObject(namespace,
Metodo2);

        // modelo del sobre
        SoapSerializationEnvelope sobre2 = new
SoapSerializationEnvelope(
            SoapEnvelope.VER11);
        sobre2.dotNet = true;
        sobre2.setOutputSoapObject(request2);

        // modelo de transporte
        HttpTransportSE transporte2 = new

HttpTransportSE(url);

        // llamada
        transporte2.call(accionSoap2, sobre2);

        // Resultado
        SoapPrimitive resultado2 = (SoapPrimitive)
sobre2.getResponse();

```

```

        texto3.setText("Centro:");
        texto4.setText("" + resultado2.toString());

    }

    catch (Exception e) {
        tv.setText(e.getMessage());
    }

    /////////////// el puente del zaragoza
    try {

        // request
        SoapObject request = new
SoapObject(namespace, Metodo3);

        // modelo del sobre
        SoapSerializationEnvelope sobre = new
SoapSerializationEnvelope(
                        SoapEnvelope.VER11);
        sobre.dotNet = true;
        sobre.setOutputSoapObject(request);

        // modelo de transporte
        HttpTransportSE transporte = new
HttpTransportSE(url);

        // llamada
        transporte.call(accionSoap3, sobre);

        // Resultado
        SoapPrimitive resultado3 = (SoapPrimitive)
sobre.getResponse();

        texto5.setText("Zaragoza:");
        texto6.setText("" + resultado3.toString());

    }

    catch (Exception e) {
        tv.setText(e.getMessage());
    }

}

    }

    /////////////// se acaba el primer boton click (el de juarez)

    if (v.getId() == findViewById(R.id.button2).getId()) {
    /////////////// empieza el boton de tijuana

        origen ="tijuana";
        origenint=2;
    }

```

```

        Vibrator vibrator =(Vibrator)
        getSystemService(Context.VIBRATOR_SERVICE);

        vibrator.vibrate(100);


        TextView texto91 = (TextView)findViewById(R.id.casapatmos);
        TextView texto101 = (TextView)findViewById(R.id.TextView01);
        TextView texto102 = (TextView)findViewById(R.id.TextView02);
        TextView texto103 = (TextView)findViewById(R.id.TextView03);
        TextView texto104 = (TextView)findViewById(R.id.TextView04);
        TextView texto105 = (TextView)findViewById(R.id.TextView05);


        TextView texto106 = (TextView)findViewById(R.id.textView4);
        TextView texto107 = (TextView)findViewById(R.id.textView5);
        TextView texto108 = (TextView)findViewById(R.id.textView6);


        texto91.setText("");
        texto101.setText("");
        texto102.setText("");
        texto103.setText("");
        texto104.setText("");
        texto105.setText("");
        texto106.setText("");
        texto107.setText("");
        texto108.setText("");


        t2.setVisibility(View.INVISIBLE);
        t3.setVisibility(View.INVISIBLE); //son la palabra "ubicacion"
        hay que volverlas visibles cuando halla casas de cambi en tijuana
        t4.setVisibility(View.INVISIBLE);


        /////el clima en tijuana
        try {

            // request
            SoapObject request6 = new

        SoapObject(namespace, Metodo6);

            // modelo del sobre
            SoapSerializationEnvelope sobre6 = new

        SoapSerializationEnvelope(

            SoapEnvelope.VER11);
            sobre6.dotNet = true;
            sobre6.setOutputSoapObject(request6);

            // modelo de transporte
            HttpTransportSE transporte6 = new

        HttpTransportSE(url);

            // llamada

```

```

        transporte6.call(accionSoap6, sobre6);

        // Resultado
        SoapPrimitive resultado6 =
(SoapPrimitive) sobre6.getResponse();

        texto7.setText("" +
resultado6.toString());

    }

    catch (Exception e) {
        tv.setText(e.getMessage());
    }

}

/////el san Ysidro
try {

    // request
    SoapObject request = new SoapObject(namespace, Metodo4);

    // modelo del sobre
    SoapSerializationEnvelope sobre = new
SoapSerializationEnvelope(
        SoapEnvelope.VER11);
    sobre.dotNet = true;
    sobre.setOutputSoapObject(request);

    // modelo de transporte
    HttpTransportSE transporte = new HttpTransportSE(url);

    // llamada
    transporte.call(accionSoap4, sobre);

    // Resultado
    SoapPrimitive resultado4 = (SoapPrimitive)
sobre.getResponse();

    texto1.setText("San Ysidro:");
    texto2.setText("" + resultado4.toString());
    texto3.setText("");

}

catch (Exception e) {
    tv.setText(e.getMessage());
}

}

```

```

    }

    if (v.getId() == findViewById(R.id.button3).getId()) {
        ////////////// empieza el boton de estadisticas

        Vibrator vibrator =(Vibrator)
        getSystemService(Context.VIBRATOR_SERVICE);

        vibrator.vibrate(100);

        if(origenint==2)
        {

            t2.setVisibility(View.INVISIBLE);
            t3.setVisibility(View.INVISIBLE); //son la palabra ubicacion hay
            que volverlas visibles cuando halla casas de cambio en tijuana
            t4.setVisibility(View.INVISIBLE);

        }

        //Toast.makeText(getApplicationContext(), "Espere
        Unos Segundos", Toast.LENGTH_LONG).show();

        Intent i = new Intent(GetServicio.this,
        Estadisticas.class);

        i.putExtra("origen", origen); // Enviando datos a
        la siguiente actividad de donde fue presionado el boton
        startActivity(i);

        }// se acaba el boton de estadisticas

    if (v.getId() == findViewById(R.id.button4).getId()) {
        ////////////// empieza el boton de alertas

        Vibrator vibrator =(Vibrator)
        getSystemService(Context.VIBRATOR_SERVICE);

        vibrator.vibrate(100);

        Toast.makeText(getApplicationContext(), "Espere
        Unos Segundos", Toast.LENGTH_LONG).show();

        Intent i = new Intent(GetServicio.this,
        Alertas.class);

        startActivity(i);

```

```

        t2.setVisibility(View.INVISIBLE);
        t3.setVisibility(View.INVISIBLE); //son la palabra
ubicacion hay que volverlas visibles cuando halla casas de cambio en tijuana
        t4.setVisibility(View.INVISIBLE);

    } // se acaba el boton de alertas

    if (v.getId() == findViewById(R.id.button5).getId()) {
        ////////////// empieza el boton de clima por hora

        Vibrator vibrator =(Vibrator)
getSystemService(Context.VIBRATOR_SERVICE);

        vibrator.vibrate(100);

        if(origenint==2)
        {

            t2.setVisibility(View.INVISIBLE);
            t3.setVisibility(View.INVISIBLE); //son la palabra ubicacion hay
que volverlas visibles cuando halla casas de cambio en tijuana
            t4.setVisibility(View.INVISIBLE);

        }

        Intent i = new Intent(GetServicio.this,
EvolucionClima.class);

        i.putExtra("origen", origen); // Enviando datos a
la siguiente actividad de donde fue presionado el boton
        startActivity(i);

    } // se acaba el boton de clima por hora

}

```

```

}

/// Pantalla Estadísticas

package com.example.webservice;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.TextView;
import android.os.Handler;
import android.content.pm.ActivityInfo;
import android.text.method.LinkMovementMethod;
import android.view.View;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapPrimitive;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

public class Estadisticas extends Activity {

    private static final String accionSoap7 =
"http://tempuri.org/estadisticascentro";
    private static final String Metodo7= "estadisticascentro";

    private static final String accionSoap8 =
"http://tempuri.org/estadisticasamericas";
    private static final String Metodo8= "estadisticasamericas";

    private static final String accionSoap9 =
"http://tempuri.org/estadisticasysleta";
    private static final String Metodo9= "estadisticasysleta";

    private static final String accionSoap10 =
"http://tempuri.org/estadisticasisidro";
    private static final String Metodo10= "estadisticasisidro";

    private static final String namespace = "http://tempuri.org/";
    private static final String url =
"http://198.216.59.184/Servicio%20Android/service1.asmx";

    public static int MILISEGUNDOS_ESPERA = 30000;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.activity_estadisticas);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_NOSENSOR);

        TextView t1 = (TextView) findViewById(R.id.textView1);
        t1.setMovementMethod(LinkMovementMethod.getInstance());

        TextView t2 = (TextView) findViewById(R.id.textView2);
        t2.setMovementMethod(LinkMovementMethod.getInstance());

        TextView t3 = (TextView) findViewById(R.id.textView3);
        t3.setMovementMethod(LinkMovementMethod.getInstance());


        Bundle origen = getIntent().getExtras();

        String a="juarez";
        String b="tijuana";
        String c="";

        c=origen.getString("origen");

        /// comparando de donde viene el click de estadisticas
        if(a.equals(c))
        {

            SoapObject request7 = new SoapObject(namespace, Metodo7);
            SoapObject request8 = new SoapObject(namespace, Metodo8);
            SoapObject request9 = new SoapObject(namespace, Metodo9);


            // modelo del sobre
            SoapSerializationEnvelope sobre7 = new
            SoapSerializationEnvelope(
                SoapEnvelope.VER11);
            sobre7.dotNet = true;
            sobre7.setOutputSoapObject(request7);

            SoapSerializationEnvelope sobre8 = new
            SoapSerializationEnvelope(
                SoapEnvelope.VER11);
            sobre8.dotNet = true;
            sobre8.setOutputSoapObject(request8);

            SoapSerializationEnvelope sobre9 = new
            SoapSerializationEnvelope(
                SoapEnvelope.VER11);
            sobre9.dotNet = true;
            sobre9.setOutputSoapObject(request9);


            // modelo de transporte
            HttpTransportSE transporte7 = new HttpTransportSE(url);
            HttpTransportSE transporte8 = new HttpTransportSE(url);
            HttpTransportSE transporte9 = new HttpTransportSE(url);

            try{

```



```

        // Resultado
        SoapPrimitive resultado10 = (SoapPrimitive)
sobre10.getResponse();

        t1.setText(resultado10.toString());
        t2.setVisibility(View.INVISIBLE);
        t3.setVisibility(View.INVISIBLE);

    }

    catch (Exception e) {
        t2.setText(e.getMessage());
    }

} /// se acaba if tijuana

        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            public void run() {
                // acciones que se ejecutan tras los milisegundos
                finish();
            }
        }, MILISEGUNDOS_ESPERA);

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_estadisticas, menu);
        return true;
    }
}

```

```

/// Pantalla Evolución por hora
package com.example.webservice;
import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;

```

```

import java.net.URL;
import java.net.URLConnection;

import android.app.Activity;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.widget.ImageView;

public class EvolucionClima extends Activity {

    private static String URLJuarez =
"http://198.216.59.184/ClimaPorHoraJuarez.jpg";
    private static String URLTijuana
="http://198.216.59.184/ClimaPorHoraTijuana.jpg";
    private static String URLFinal="";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.evolucionclima);

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_NOSENSOR);

        Bundle origen = getIntent().getExtras();

        String a="juarez";
        String b="tijuana";
        String c="";

        c=origen.getString("origen");

        //// comparando de donde viene el click del clima
        if(a.equals(c))
        {
            URLFinal=URLJuarez; // es de Juarez el Clima
        }
        else
        {
            URLFinal=URLTijuana; // es de Tijuana el Clima
        }

        Bitmap bitmap = DownloadImage(URLFinal);

        ImageView img = (ImageView) findViewById(R.id.img);
        img.setImageBitmap(bitmap);
    }
}

```

```

} // del onCreate

private InputStream OpenHttpConnection(String urlString)
    throws IOException
{
    InputStream in = null;
    int response = -1;

    URL url = new URL(urlString);
    URLConnection conn = url.openConnection();

    if (!(conn instanceof HttpURLConnection))
        throw new IOException("Not an HTTP connection");

    try{
        HttpURLConnection httpConn = (HttpURLConnection)
conn;

        httpConn.setAllowUserInteraction(false);
        httpConn.setInstanceFollowRedirects(true);
        httpConn.setRequestMethod("GET");
        httpConn.connect();

        response = httpConn.getResponseCode();
        if (response == HttpURLConnection.HTTP_OK) {
            in = httpConn.getInputStream();
        }
    }
    catch (Exception ex)
    {
        throw new IOException("Error connecting");
    }
    return in;
}

private Bitmap DownloadImage(String URL)
{
    Bitmap bitmap = null;
    InputStream in = null;
    try {
        in = OpenHttpConnection(URL);
        bitmap = BitmapFactory.decodeStream(in);
        in.close();
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
    return bitmap;
}

} // del extends activity

```

```

/// Pantalla Alertas
package com.example.webservice;

import org.ksoap2.SoapEnvelope;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapPrimitive;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;


import android.app.Activity;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;

import android.content.Context;
import android.content.Intent;
import android.content.pm.ActivityInfo;


import android.os.Bundle;
import android.os.Handler;


import android.view.View;
import android.widget.TextView;
import android.widget.Toast;


public class Alertas extends Activity implements View.OnClickListener {

    private static final String accionSoap1 =
"http://tempuri.org/alertaamericas";
    private static final String accionSoap2 =
"http://tempuri.org/alertacentro";
    private static final String accionSoap3 =
"http://tempuri.org/alertaysleta";
    private static final String accionSoap4 =
"http://tempuri.org/alertaisidro";
    private static final String Metodo1= "alertaamericas";
    private static final String Metodo2= "alertacentro";
    private static final String Metodo3= "alertaysleta";
    private static final String Metodo4= "alertaisidro";
    private static final String namespace = "http://tempuri.org/";
    private static final String url =
"http://198.216.59.184/Servicio%20Android/service1.asmx";

```

```

        public static int MILISEGUNDOS_ESPERA = 5000; ///moverlo a 15
minutos...
        public static int estadoalerta = 16; // 16 veces de 15 minutos... 4
hrs checando alertas

```

```

/** Called when the activity is first created. */
@Override

```

```

    public void onCreate(Bundle savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);

```

```

        setContentView(R.layout.alertas);

```

```

        View botonocultar = findViewById(R.id.button1);
        botonocultar.setOnClickListener(this);
        View boton1 = findViewById(R.id.boton1);
        boton1.setOnClickListener(this);

```

```

        View boton2 = findViewById(R.id.boton2);
        boton2.setOnClickListener(this);

```

```

        View boton3 = findViewById(R.id.Button01);
        boton3.setOnClickListener(this);

```

```

        View boton4 = findViewById(R.id.boton4);
        boton4.setOnClickListener(this);

```

```

        View boton5 = findViewById(R.id.Button05);
        boton5.setOnClickListener(this);

```

```

        View boton6 = findViewById(R.id.boton6);
        boton6.setOnClickListener(this);

```

```

        View boton7 = findViewById(R.id.boton7);
        boton7.setOnClickListener(this);

```

```

        View boton8 = findViewById(R.id.boton8);
        boton8.setOnClickListener(this);

```

```

        View boton9 = findViewById(R.id.boton9);
        boton9.setOnClickListener(this);

```

```

        View boton10 = findViewById(R.id.boton10);
        boton10.setOnClickListener(this);

```

```

        View boton11 = findViewById(R.id.boton11);
        boton11.setOnClickListener(this);

```

```

        View boton12 = findViewById(R.id.boton12);
        boton12.setOnClickListener(this);

```

```

setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_NOSENSOR);

}

@SuppressWarnings("deprecation")
@Override
public void onClick(View v) {
    // TODO Auto-generated method stub

    final TextView texto2 = (TextView)findViewById(R.id.textView2);
    texto2.setText("");

    if (v.getId() == findViewById(R.id.button1).getId()) {
        ////////// boton ocultar

        this.moveTaskToBack(true);
    } //final del boton ocultar app

    if (v.getId() == findViewById(R.id.boton1).getId()) {
        ////////// boton 20 min americas

        final NotificationManager mNotificationManager =
            (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

        //Agregando el icono, texto y momento para lanzar
la notificación

        int icon = R.drawable.icono_azul;
        CharSequence tickerText = "Notification Bar";
        long when = System.currentTimeMillis();

        final Notification notification = new
Notification(icon, tickerText, when);

        Context context = getApplicationContext();
        CharSequence contentType = "Tiempo de Puentes";
        CharSequence contentText = "¡El tiempo de Cruce
esta como lo deseas!";

        //Agregando sonido
notification.defaults |=
Notification.DEFAULT_SOUND;

```

```

        //Agregando vibración
        notification.defaults |=
Notification.DEFAULT_VIBRATE;
        notification.flags |=
Notification.FLAG_AUTO_CANCEL;

        Intent notificationIntent = new Intent(this,
Alertas.class);
        PendingIntent contentIntent =
PendingIntent.getActivity(this, 0, notificationIntent, 0);
        notification.setLatestEventInfo(context,
contentTitle, contentText, contentIntent);

        //Toast.makeText(getApplicationContext(), "Espere Unos
Segundos", Toast.LENGTH_LONG).show();
        Toast.makeText(getApplicationContext(), "Alerta Activada
para 20 Minutos", Toast.LENGTH_LONG).show();
        texto2.setText("20 Minutos para Puente Américas");

        estadoalerta=16;

        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            public void run() {
                // acciones que se ejecutan tras los milisegundos

                while(estadoalerta>0)
                {

                    try {

                        // request
                        SoapObject request1 = new
SoapObject(namespace, Metodo1);

                        // modelo del sobre
                        SoapSerializationEnvelope sobre1 = new
SoapSerializationEnvelope(
                            SoapEnvelope.VER11);
                        sobre1.dotNet = true;
                        sobre1.setOutputSoapObject(request1);
                        // modelo de transporte
                        HttpTransportSE transporte1 = new
HttpTransportSE(url);

                        // llamada
                        transporte1.call(accionSoap1, sobre1);
                        // Resultado
                        SoapPrimitive resultado1 = (SoapPrimitive)
sobre1.getResponse();

                        String resultado11 = resultado1.toString();
                        int tiempoentero =

                        if(tiempoentero<25&&tiempoentero!=0)
                        {

```



```

        estadoalerta=0;

        mNotificationManager.notify(101,
notification); /// se lanza la notificación

        Intent i=new
Intent(Alertas.this,AlertaActivada.class);

i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);///optional

i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);///will cause it to come to
foreground
        i.putExtra("resultado11", resultado11); //
Enviando datos a la siguiente actividad de donde fue presionado el boton
startActivity(i);

        texto2.setText("¡¡El tiempo actual es
de "+resultado11.toString()+" minutos!!");

        finish();

    }

    } catch (Exception e) {
        texto2.setText(e.getMessage());
    }

        estadoalerta--; // llega a 0 despues de 4
hrs de estar checando cada 15 minutos

    }/// del while

    }

    private void moveTaskToBack(boolean b) {
        // TODO Auto-generated method stub

    }

}, MILISEGUNDOS_ESPERA);

}///final del boton 20 de puente americas

```

```

        if (v.getId() == findViewById(R.id.boton2).getId()) {
            ////////////////////////////////////////////////// boton 40 min americas

            final NotificationManager mNotificationManager =
                (NotificationManager)
                getSystemService(Context.NOTIFICATION_SERVICE);

            //Agregando el icono, texto y momento para lanzar
            la notificación

            int icon = R.drawable.icono_azul;
            CharSequence tickerText = "Notification Bar";
            long when = System.currentTimeMillis();

            final Notification notification = new
            Notification(icon, tickerText, when);

            Context context = getApplicationContext();
            CharSequence contentType = "Tiempo de Puentes";
            CharSequence contentText = "¡El tiempo de Cruce
            esta como lo deseas!";

            //Agregando sonido
            notification.defaults |=
            Notification.DEFAULT_SOUND;
            //Agregando vibración
            notification.defaults |=
            Notification.DEFAULT_VIBRATE;
            notification.flags |=
            Notification.FLAG_AUTO_CANCEL;

            Intent notificationIntent = new Intent(this,
            Alertas.class);

            PendingIntent contentIntent =
            PendingIntent.getActivity(this, 0, notificationIntent, 0);
            notification.setLatestEventInfo(context,
            contentType, contentText, contentIntent);

            //Toast.makeText(getApplicationContext(), "Espere Unos
            Segundos", Toast.LENGTH_LONG).show();
            Toast.makeText(getApplicationContext(), "Alerta Activada
            para 40 Minutos", Toast.LENGTH_LONG).show();
            texto2.setText("40 Minutos para Puente Américas");

            estadoalerta=16;

            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                public void run() {
                    // acciones que se ejecutan tras los milisegundos

                    while(estadoalerta>0)
                    {

                        try {

                            // request

```

```

SoapObject request1 = new
SoapObject(namespace, Metodo1);

// modelo del sobre
SoapSerializationEnvelope sobre1 = new
SoapSerializationEnvelope(
    SoapEnvelope.VER11);
sobre1.dotNet = true;
sobre1.setOutputSoapObject(request1);
// modelo de transporte
HttpTransportSE transporte1 = new
HttpTransportSE(url);

// llamada
transporte1.call(accionSoap1, sobre1);
// Resultado
SoapPrimitive resultado1 = (SoapPrimitive)
sobre1.getResponse();

String resultado11 = resultado1.toString();
int tiempoentero =

Integer.parseInt(resultado11);

if(tiempoentero<45&&tiempoentero!=0)
{

    estadoalerta=0;

    mNotificationManager.notify(101,
notification); /// se lanza la notificación

Intent i=new
Intent(Alertas.this,AlertaActivada.class);

i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
foreground
i.putExtra("resultado11", resultado11); //
Enviando datos a la siguiente actividad de donde fue presionado el boton
startActivity(i);

texto2.setText("¡¡El tiempo actual es
de "+resultado11.toString()+" minutos!!");

finish();

```

```

    }

    } catch (Exception e) {
        texto2.setText(e.getMessage());
    }

        estadoalerta--; // llega a 0 despues de 4
hrs de estar checando cada 15 minutos

    } // del while
}

    private void moveTaskToBack(boolean b) {
        // TODO Auto-generated method stub
    }
}, MILISEGUNDOS_ESPERA);

} // final del boton 40 de puente americas

if (v.getId() == findViewById(R.id.Button01).getId()) {
    // boton 60 min americas

    final NotificationManager
mNotificationManager =
                                (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

    // Agregando el icono, texto y momento
para lanzar la notificación

    int icon = R.drawable.icono_azul;
    CharSequence tickerText =
"Notification Bar";
    long when =
System.currentTimeMillis();

    final Notification notification = new
Notification(icon, tickerText, when);

    Context context =
getApplicationContext();
    CharSequence contentTitle = "Tiempo de
Puentes";
    CharSequence contentText = "¡El tiempo
de Cruce esta como lo deseas!";

    // Agregando sonido
notification.defaults |=
Notification.DEFAULT_SOUND;

```

```

//Agregando vibración
notification.defaults |=
notification.flags |=

Notification.DEFAULT_VIBRATE;
Notification.FLAG_AUTO_CANCEL;

Intent notificationIntent = new
PendingIntent getActivity(this, 0, notificationIntent, 0);

notification.setLatestEventInfo(context, contentTitle, contentText,
contentIntent);

//Toast.makeText(getApplicationContext(),
"Espere Unos Segundos", Toast.LENGTH_LONG).show();
Toast.makeText(getApplicationContext(),
"Alerta Activada para 60 Minutos", Toast.LENGTH_LONG).show();
texto2.setText("60 Minutos para Puente
Américas");

estadoalerta=16;

Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    public void run() {
        // acciones que se ejecutan tras los
        milisegundos

        while(estadoalerta>0)
        {

            try {

                // request
                SoapObject request1 = new

                SoapObject(namespace, Metodo1);

                // modelo del sobre
                SoapSerializationEnvelope sobre1

                = new SoapSerializationEnvelope(
                    SoapEnvelope.VER11);

                sobre1.dotNet = true;

                sobre1.setOutputSoapObject(request1);

                // modelo de transporte
                HttpTransportSE transporte1 =

                // llamada
                transporte1.call(accionSoap1,
                sobre1);

                // Resultado
                SoapPrimitive resultado1 =

                String resultado11 =

                resultado1.toString();

```

```

Integer.parseInt(resultado11);

        int tiempoentero =

        if(tiempoentero<65&&tiempoentero!=0)
        {

            estadoalerta=0;

            mNotificationManager.notify(101, notification); /// se lanza la
            notificación

            Intent i=new
            Intent(Alertas.this,AlertaActivada.class);

            i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

            i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
            foreground

            i.putExtra("resultado11",
            resultado11); // Enviando datos a la siguiente actividad de donde fue
            presionado el boton

            startActivity(i);

            texto2.setText("¡¡El
            tiempo actual es de "+resultado11.toString()+" minutos!!");

            finish();

        }

    } catch (Exception e) {
        texto2.setText(e.getMessage());

    }

    estadoalerta--; // llega a 0
    despues de 4 hrs de estar checando cada 15 minutos

    }// del while

}

```



```

        notification.setLatestEventInfo(context, contentTitle, contentText,
contentIntent);

        //Toast.makeText(getApplicationContext(),
"Espere Unos Segundos", Toast.LENGTH_LONG).show();
        Toast.makeText(getApplicationContext(),
"Alerta Activada para 120 Minutos", Toast.LENGTH_LONG).show();
        texto2.setText("120 Minutos para Puente
Américas");

        estadoalerta=16;

        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            public void run() {
                // acciones que se ejecutan tras los
milisegundos

                while(estadoalerta>0)
                {

                    try {

                        // request
                        SoapObject request1 = new

SoapObject(namespace, Metodo1);

                        // modelo del sobre
                        SoapSerializationEnvelope sobre1

= new SoapSerializationEnvelope(

                        SoapEnvelope.VER11);

                        sobre1.dotNet = true;

                        sobre1.setOutputSoapObject(request1);

                        // modelo de transporte
                        HttpTransportSE transporte1 =

                        // llamada
                        transporte1.call(accionSoap1,

                        // Resultado
                        SoapPrimitive resultado1 =

                        String resultado11 =

                        int tiempoentero =

                        if(tiempoentero<125&&tiempoentero!=0)
                    {

```



```

                                estadoalerta=0;

                                mNotificationManager.notify(101, notification); /// se lanza la
notificación

                                Intent i=new
Intent(Alertas.this,AlertaActivada.class);

                                i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

                                i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
foreground

                                i.putExtra("resultado11",
resultado11); // Enviando datos a la siguiente actividad de donde fue
presionado el boton

                                startActivity(i);

                                texto2.setText("¡¡El
tiempo actual es de "+resultado11.toString()+" minutos!!");

                                finish();

                                }

                                } catch (Exception e) {
                                texto2.setText(e.getMessage());

                                }

                                estadoalerta--; // llega a 0
despues de 4 hrs de estar checando cada 15 minutos

                                }// del while

                                }

                                private void moveTaskToBack(boolean b)
{
                                // TODO Auto-generated method
stub

                                }
}, MILISEGUNDOS_ESPERA);

```

```

        }//final del boton 120 de puente americas

        if (v.getId() == findViewById(R.id.Button05).getId()) {
            /////////// boton 20 min centro

            final NotificationManager
mNotificationManager =
                                (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

            //Agregando el icono, texto y momento
para lanzar la notificación
            int icon = R.drawable.icono_azul;
            CharSequence tickerText =
"Notification Bar";
            long when =
System.currentTimeMillis();

            final Notification notification = new
Notification(icon, tickerText, when);

            Context context =
getApplicationContext();
            CharSequence contentTitle = "Tiempo de
Puentes";
            CharSequence contentText = "¡El tiempo
de Cruce esta como lo deseas!";

            //Agregando sonido
notification.defaults |=
            //Agregando vibración
notification.defaults |=
            notification.flags |=

            Intent notificationIntent = new
PendingIntent.getActivity(this, 0, notificationIntent, 0);

            notification.setLatestEventInfo(context, contentTitle, contentText,
contentIntent);

            //Toast.makeText(getApplicationContext(),
"Espere Unos Segundos", Toast.LENGTH_LONG).show();
            Toast.makeText(getApplicationContext(),
"Alerta Activada para 20 Minutos", Toast.LENGTH_LONG).show();
            texto2.setText("20 Minutos para Puente
Centro");

```

```

        estadoalerta=16;

        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            public void run() {
                // acciones que se ejecutan tras los
milisegundos

                while(estadoalerta>0)
                {

                    try {

                        // request
                        SoapObject request2 = new

SoapObject(namespace, Metodo2);

                        // modelo del sobre
                        SoapSerializationEnvelope sobre2

= new SoapSerializationEnvelope(

                        SoapEnvelope.VER11);

                        sobre2.setOutputSoapObject(request2);

                        // modelo de transporte
                        HttpTransportSE transporte2 =

new HttpTransportSE(url);

                        // llamada
                        transporte2.call(accionSoap2,

sobre2);

                        // Resultado
                        SoapPrimitive resultado1 =

(SoapPrimitive) sobre2.getResponse();

                        String resultado11 =

resultado1.toString();

                        int tiempoentero =

Integer.parseInt(resultado11);

                        if(tiempoentero<25&&tiempoentero!=0)

                        {

                                estadoalerta=0;

                                mNotificationManager.notify(101, notification); /// se lanza la
notificación

```

```

Intent i=new
Intent(Alertas.this,AlertaActivada.class);

i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
foreground
i.putExtra("resultado11",
resultado11); // Enviando datos a la siguiente actividad de donde fue
presionado el boton
startActivity(i);

texto2.setText("¡¡El
tiempo actual es de "+resultado11.toString()+" minutos!!");

finish();

}

} catch (Exception e) {
texto2.setText(e.getMessage());
}

estadoalerta--; // llega a 0
despues de 4 hrs de estar checando cada 15 minutos

} // del while

}

private void moveTaskToBack(boolean b)
{
// TODO Auto-generated method
stub

}

}, MILISEGUNDOS_ESPERA);

} //final del boton 20 de puente centro

if (v.getId() == findViewById(R.id.boton6).getId()) {
////////// boton 40 min centro

final NotificationManager
mNotificationManager =

```

```

        (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);

        //Agregando el icono, texto y momento
        para lanzar la notificación
        int icon = R.drawable.icono_azul;
        CharSequence tickerText =

        "Notification Bar";
        long when =

        System.currentTimeMillis();

        final Notification notification = new
        Notification(icon, tickerText, when);

        Context context =

        getApplicationContext();
        CharSequence contentTitle = "Tiempo de
        Puentes";
        CharSequence contentText = "¡El tiempo
        de Cruce esta como lo deseas!";

        //Agregando sonido
        notification.defaults |=

        //Agregando vibración
        notification.defaults |=

        notification.flags |=

        Intent notificationIntent = new

        Intent(this, Alertas.class);
        PendingIntent contentIntent =

        PendingIntent.getActivity(this, 0, notificationIntent, 0);

        notification.setLatestEventInfo(context, contentTitle, contentText,
        contentIntent);

        //Toast.makeText(getApplicationContext(),
        "Espere Unos Segundos", Toast.LENGTH_LONG).show();
        Toast.makeText(getApplicationContext(),
        "Alerta Activada para 40 Minutos", Toast.LENGTH_LONG).show();
        texto2.setText("40 Minutos para Puente
        Centro");

        estadoalerta=16;

        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            public void run() {
                // acciones que se ejecutan tras los
                milisegundos

                while(estadoalerta>0)
                {

                    try {

```

```

SoapObject(namespace, Metodo2);

= new SoapSerializationEnvelope(
    SoapEnvelope.VER11);

    sobre2.setOutputSoapObject(request2);

new HttpTransportSE(url);

sobre2);

(SoapPrimitive) sobre2.getResponse();

resultado1.toString();

Integer.parseInt(resultado11);

        if(tiempoentero<45&&tiempoentero!=0)
        {

                estadoalerta=0;

        mNotificationManager.notify(101, notification); /// se lanza la
notificación

        Intent i=new
Intent(Alertas.this,AlertaActivada.class);

i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
foreground

                i.putExtra("resultado11",
resultado11); // Enviando datos a la siguiente actividad de donde fue
presionado el boton

                startActivity(i);

```

```

                                texto2.setText("¡¡El
tiempo actual es de "+resultado11.toString()+" minutos!!");

                                finish();
                                }

                                } catch (Exception e) {
                                    texto2.setText(e.getMessage());
                                }

                                estadoalerta--; // llega a 0
despues de 4 hrs de estar checando cada 15 minutos
                                }// del while
                                }

                                private void moveTaskToBack(boolean b)
{
                                // TODO Auto-generated method
                                stub

                                }
                                }, MILISEGUNDOS_ESPERA);

                                }//final del boton 40 de puente centro

                                if (v.getId() == findViewById(R.id.boton7).getId()) {
                                    /////////// boton 60 min centro

                                final NotificationManager
mNotificationManager =
                                (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

                                //Agregando el icono, texto y momento
para lanzar la notificación
                                int icon = R.drawable.icono_azul;
                                CharSequence tickerText =
                                "Notification Bar";
                                long when =
                                System.currentTimeMillis();

                                final Notification notification = new
Notification(icon, tickerText, when);

```

```

getApplicationContext();
Puentes";
de Cruce esta como lo deseas!";

Notification.DEFAULT_SOUND;
Notification.DEFAULT_VIBRATE;
Notification.FLAG_AUTO_CANCEL;

Intent(this, Alertas.class);
PendingIntent.getActivity(this, 0, notificationIntent, 0);

notification.setLatestEventInfo(context, contentTitle, contentText,
contentIntent);

//Toast.makeText(getApplicationContext(),
"Espere Unos Segundos", Toast.LENGTH_LONG).show();
Toast.makeText(getApplicationContext(),
"Alerta Activada para 60 Minutos", Toast.LENGTH_LONG).show();
texto2.setText("60 Minutos para Puente
Centro");

estadoalerta=16;

Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    public void run() {
        // acciones que se ejecutan tras los
milisegundos

        while(estadoalerta>0)
        {

            try {

                // request
                SoapObject request2 = new

                // modelo del sobre
                SoapSerializationEnvelope sobre2

                sobre2.dotNet = true;

                sobre2.setOutputSoapObject(request2);
                // modelo de transporte

```



```

new HttpTransportSE(url);

sobre2);

(SoapPrimitive) sobre2.getResponse();

resultado1.toString();

Integer.parseInt(resultado11);

HttpTransportSE transporte2 =
// llamada
transporte2.call(accionSoap2,

// Resultado
SoapPrimitive resultado1 =

String resultado11 =
int tiempoentero =

if(tiempoentero<65&&tiempoentero!=0)
{

    estadoalerta=0;

    mNotificationManager.notify(101, notification); /// se lanza la
    notificación

    Intent i=new
    Intent(Alertas.this,AlertaActivada.class);

    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

    i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
    foreground

    i.putExtra("resultado11",
    resultado11); // Enviando datos a la siguiente actividad de donde fue
    presionado el boton

    startActivity(i);

    texto2.setText("¡¡El
    tiempo actual es de "+resultado11.toString()+" minutos!!");

    finish();

}

} catch (Exception e) {

```

```

        texto2.setText(e.getMessage());
    }

    estadoalerta--; // llega a 0
    despues de 4 hrs de estar checando cada 15 minutos

} // del while

}

private void moveTaskToBack(boolean b)
{
    // TODO Auto-generated method stub

}

}, MILISEGUNDOS_ESPERA);

} // final del boton 60 de puente centro

if (v.getId() == findViewById(R.id.boton8).getId()) {
    //////////// boton 120 min centro

final NotificationManager
mNotificationManager =
(NotificationManager)
getService(Context.NOTIFICATION_SERVICE);

//Agregando el icono, texto y momento
para lanzar la notificación

int icon = R.drawable.icono_azul;
CharSequence tickerText =

"Notification Bar";

long when =

System.currentTimeMillis();

final Notification notification = new

Notification(icon, tickerText, when);

Context context =

CharSequence contentTypeTitle = "Tiempo de

CharSequence contentTypeText = "¡El tiempo

de Cruce esta como lo deseas!";

//Agregando sonido
notification.defaults |=

//Agregando vibración
notification.defaults |=

```

```

        notification.flags |=
Notification.FLAG_AUTO_CANCEL;

        Intent notificationIntent = new
Intent(this, Alertas.class);
        PendingIntent contentIntent =
PendingIntent.getActivity(this, 0, notificationIntent, 0);

        notification.setLatestEventInfo(context, contentTitle, contentText,
contentIntent);

        //Toast.makeText(getApplicationContext(),
"Espere Unos Segundos", Toast.LENGTH_LONG).show();
        Toast.makeText(getApplicationContext(),
"Alerta Activada para 120 Minutos", Toast.LENGTH_LONG).show();
        texto2.setText("120 Minutos para Puente
Centro");

        estadoalerta=16;

        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            public void run() {
                // acciones que se ejecutan tras los
milisegundos

                while(estadoalerta>0)
                {

                    try {

                        // request
                        SoapObject request2 = new

                        // modelo del sobre
                        SoapSerializationEnvelope sobre2

= new SoapSerializationEnvelope(
                        SoapEnvelope.VER11);

                        sobre2.setOutputSoapObject(request2);

                        // modelo de transporte
                        HttpTransportSE transporte2 =

                        // llamada
                        transporte2.call(accionSoap2,

                        // Resultado
                        SoapPrimitive resultado1 =

                        String resultado11 =

                        int tiempoentero =

```

```

        if(tiempoentero<125&&tiempoentero!=0)
        {

            estadoalerta=0;

            mNotificationManager.notify(101, notification); /// se lanza la
            notificación

            Intent i=new
            Intent(Alertas.this,AlertaActivada.class);

            i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

            i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
            foreground

            i.putExtra("resultado11",
            resultado11); // Enviando datos a la siguiente actividad de donde fue
            presionado el boton

            startActivity(i);

            texto2.setText("¡¡El
            tiempo actual es de "+resultado11.toString()+" minutos!!");

            finish();

        }

    } catch (Exception e) {
        texto2.setText(e.getMessage());

    }

    estadoalerta--; // llega a 0
    despues de 4 hrs de estar checando cada 15 minutos

    }// del while

    }

    private void moveTaskToBack(boolean b)
{

```

```

// TODO Auto-generated method
stub

    }
    }, MILISEGUNDOS_ESPERA);

}

} //final del boton 120 de puente centro

if (v.getId() == findViewById(R.id.boton9).getId()) {
    //boton 20 min Ysleta

    final NotificationManager
mNotificationManager =
        (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);

    //Agregando el icono, texto y momento
    para lanzar la notificación
    int icon = R.drawable.icono_azul;
    CharSequence tickerText =
    "Notification Bar";
    long when =
    System.currentTimeMillis();

    final Notification notification = new
    Notification(icon, tickerText, when);

    Context context =
    getApplicationContext();
    CharSequence contentTitle = "Tiempo de
    Puentes";
    CharSequence contentText = "¡El tiempo
    de Cruce esta como lo deseas!";

    //Agregando sonido
    notification.defaults |=
    //Agregando vibración
    notification.defaults |=
    notification.flags |=

    Intent notificationIntent = new
    Intent(this, Alertas.class);
    PendingIntent contentIntent =
    PendingIntent.getActivity(this, 0, notificationIntent, 0);

    notification.setLatestEventInfo(context, contentTitle, contentText,
    contentIntent);

```

```

//Toast.makeText(getApplicationContext(),
"Espere Unos Segundos", Toast.LENGTH_LONG).show();
Toast.makeText(getApplicationContext(),
"Alerta Activada para 20 Minutos", Toast.LENGTH_LONG).show();
texto2.setText("20 Minutos para Puente
Zaragoza");

estadoalerta=16;

Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    public void run() {
        // acciones que se ejecutan tras los
milisegundos

        while(estadoalerta>0)
        {

            try {

                // request
                SoapObject request3 = new

                SoapObject(namespace, Metodo3);

                // modelo del sobre
                SoapSerializationEnvelope sobre3

                = new SoapSerializationEnvelope(

                    SoapEnvelope.VER11);

                sobre3.setOutputSoapObject(request3);

                sobre3.dotNet = true;

                // modelo de transporte
                HttpTransportSE transporte3 =

                new HttpTransportSE(url);

                // llamada
                transporte3.call(accionSoap3,

                sobre3);

                // Resultado
                SoapPrimitive resultado1 =

                (SoapPrimitive) sobre3.getResponse();

                String resultado11 =

                resultado1.toString();

                int tiempoentero =

                Integer.parseInt(resultado11);

                if(tiempoentero<25&&tiempoentero!=0)
                {

                    estadoalerta=0;

                    mNotificationManager.notify(101, notification); /// se lanza la
notificación

```

```

Intent i=new
Intent(Alertas.this,AlertaActivada.class);

i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
foreground
i.putExtra("resultado11",
resultado11); // Enviando datos a la siguiente actividad de donde fue
presionado el boton
startActivity(i);

texto2.setText("¡¡El
tiempo actual es de "+resultado11.toString()+" minutos!!");

finish();

}

} catch (Exception e) {
texto2.setText(e.getMessage());
}

estadoalerta--; // llega a 0
despues de 4 hrs de estar checando cada 15 minutos
} // del while
}

private void moveTaskToBack(boolean b)
{
// TODO Auto-generated method
stub

}
}, MILISEGUNDOS_ESPERA);

} //final del boton 20 de puente Ysleta

```

```

        if (v.getId() == findViewById(R.id.boton10).getId()) {
            ////////////////////////////////////////////////// boton 40 min Ysleta

            final NotificationManager
mNotificationManager =
                                (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

            //Agregando el icono, texto y momento
para lanzar la notificación
            int icon = R.drawable.icono_azul;
            CharSequence tickerText =
"Notification Bar";
            long when =
System.currentTimeMillis();

            final Notification notification = new
Notification(icon, tickerText, when);

            Context context =
getApplicationContext();
            CharSequence contentType = "Tiempo de
Puentes";
            CharSequence contentText = "¡El tiempo
de Cruce esta como lo deseas!";

            //Agregando sonido
notification.defaults |=
Notification.DEFAULT_SOUND;

            //Agregando vibración
notification.defaults |=
Notification.DEFAULT_VIBRATE;

            notification.flags |=
Notification.FLAG_AUTO_CANCEL;

            Intent notificationIntent = new
Intent(this, Alertas.class);
            PendingIntent contentIntent =
PendingIntent.getActivity(this, 0, notificationIntent, 0);

            notification.setLatestEventInfo(context, contentType, contentText,
contentIntent);

            //Toast.makeText(getApplicationContext(),
"Espere Unos Segundos", Toast.LENGTH_LONG).show();
            Toast.makeText(getApplicationContext(),
"Alerta Activada para 40 Minutos", Toast.LENGTH_LONG).show();
            texto2.setText("40 Minutos para Puente
Zaragoza");

            estadoalerta=16;

            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                public void run() {

```



```

// acciones que se ejecutan tras los
milisegundos

while(estadoalerta>0)
{
    try {
        // request
        SoapObject request3 = new
SoapObject(namespace, Metodo3);

        // modelo del sobre
        SoapSerializationEnvelope sobre3
= new SoapSerializationEnvelope(
        SoapEnvelope.VER11);

        sobre3.setOutputSoapObject(request3);

        // modelo de transporte
        HttpTransportSE transporte3 =
        // llamada
        transporte3.call(accionSoap3,
        sobre3);

        // Resultado
        SoapPrimitive resultado1 =
        (SoapPrimitive) sobre3.getResponse();

        String resultado11 =
        resultado1.toString();

        int tiempoentero =
        Integer.parseInt(resultado11);

        if(tiempoentero<45&&tiempoentero!=0)
        {

            estadoalerta=0;

            mNotificationManager.notify(101, notification); /// se lanza la
notificación

            Intent i=new
            Intent(Alertas.this,AlertaActivada.class);

            i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

```

```

i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
foreground
                                i.putExtra("resultado11",
resultado11); // Enviando datos a la siguiente actividad de donde fue
presionado el boton
                                startActivity(i);

                                texto2.setText("¡¡El
tiempo actual es de "+resultado11.toString()+" minutos!!");

                                finish();

                                }

                                } catch (Exception e) {
                                    texto2.setText(e.getMessage());
                                }

                                estadoalerta--; // llega a 0
despues de 4 hrs de estar checando cada 15 minutos
                                }// del while
                                }

                                private void moveTaskToBack(boolean b)
{
                                // TODO Auto-generated method
stub

                                }
                                }, MILISEGUNDOS_ESPERA);

                                }//final del boton 40 de puente Ysleta

                                if (v.getId() == findViewById(R.id.boton11).getId()) {
                                    /////////// boton 60 min Ysleta

                                final NotificationManager
mNotificationManager =
                                (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

                                //Agregando el icono, texto y momento
para lanzar la notificación

```

```

        int icon = R.drawable.icono_azul;
        CharSequence tickerText =

"Notification Bar";

        System.currentTimeMillis();

        Notification(notification, icon, tickerText, when);

        Context context =

        CharSequence contentType = "Tiempo de

        CharSequence contentType = "¡El tiempo

        //Agregando sonido
        notification.defaults |=

        //Agregando vibración
        notification.defaults |=

        notification.flags |=

        Intent notificationIntent = new

        PendingIntent contentIntent =

        PendingIntent.getActivity(this, 0, notificationIntent, 0);

        notification.setLatestEventInfo(context, contentType, contentType,
        contentIntent);

        //Toast.makeText(getApplicationContext(),
        "Espere Unos Segundos", Toast.LENGTH_LONG).show();
        Toast.makeText(getApplicationContext(),
        "Alerta Activada para 60 Minutos", Toast.LENGTH_LONG).show();
        texto2.setText("60 Minutos para Puente

        Zaragoza");

        estadoalerta=16;

        Handler handler = new Handler();
        handler.postDelayed(new Runnable() {
            public void run() {
                // acciones que se ejecutan tras los

                milisegundos

                while(estadoalerta>0)
                {

                    try {

                        // request
                        SoapObject request3 = new

                        // modelo del sobre

```

```

= new SoapSerializationEnvelope(
    SoapEnvelope.VER11);

    sobre3.setOutputSoapObject(request3);

new HttpTransportSE(url);

sobre3);

(SoapPrimitive) sobre3.getResponse();

resultado1.toString();

Integer.parseInt(resultado11);

    if(tiempoentero<65&&tiempoentero!=0)
    {

        estadoalerta=0;

        mNotificationManager.notify(101, notification); /// se lanza la
        notificación

        Intent i=new
        Intent(Alertas.this,AlertaActivada.class);

        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

        i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
        foreground

        i.putExtra("resultado11",
        resultado11); // Enviando datos a la siguiente actividad de donde fue
        presionado el boton

        startActivity(i);

        texto2.setText("¡¡El
        tiempo actual es de "+resultado11.toString()+" minutos!!");

```

```

        finish();
    }

    } catch (Exception e) {
        texto2.setText(e.getMessage());
    }

    estadoalerta--; // llega a 0
    despues de 4 hrs de estar checando cada 15 minutos

    }// del while

    }

    private void moveTaskToBack(boolean b)
    {
        // TODO Auto-generated method
        stub

    }
    }, MILISEGUNDOS_ESPERA);

    }//final del boton 60 de puente Ysleta

    if (v.getId() == findViewById(R.id.boton12).getId()) {
        /////////// boton 120 min Ysleta

        final NotificationManager
mNotificationManager =
        (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);

        //Agregando el icono, texto y momento
        para lanzar la notificación

        int icon = R.drawable.icono_azul;
        CharSequence tickerText =
        "Notification Bar";
        long when =
        System.currentTimeMillis();

        final Notification notification = new
Notification(icon, tickerText, when);

        Context context =
        CharSequence contentTitle = "Tiempo de
Puentes";

```

```

CharSequence contentText = "¡El tiempo
de Cruce esta como lo deseas!";

//Agregando sonido
notification.defaults |=
Notification.DEFAULT_SOUND;

//Agregando vibración
notification.defaults |=
Notification.DEFAULT_VIBRATE;

notification.flags |=
Notification.FLAG_AUTO_CANCEL;

Intent notificationIntent = new
PendingIntent.getActivity(this, 0, notificationIntent, 0);

notification.setLatestEventInfo(context, contentTitle, contentText,
contentIntent);

//Toast.makeText(getApplicationContext(),
"Espere Unos Segundos", Toast.LENGTH_LONG).show();
Toast.makeText(getApplicationContext(),
"Alerta Activada para 120 Minutos", Toast.LENGTH_LONG).show();
texto2.setText("120 Minutos para Puente
Zaragoza");

estadoalerta=16;

Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    public void run() {
        // acciones que se ejecutan tras los
        milisegundos

        while(estadoalerta>0)
        {

            try {

                // request
                SoapObject request3 = new

                // modelo del sobre
                SoapSerializationEnvelope sobre3

                = new SoapSerializationEnvelope(
                    SoapEnvelope.VER11);

                sobre3.setOutputSoapObject(request3);

                // modelo de transporte
                HttpTransportSE transporte3 =

                // llamada
                transporte3.call(accionSoap3,
                sobre3);

```

```

// Resultado
SoapPrimitive resultado1 =

(SoapPrimitive) sobre3.getResponse();

resultado1.toString();

Integer.parseInt(resultado1);

    if(tiempoentero<125&&tiempoentero!=0)
    {

        estadoalerta=0;

        mNotificationManager.notify(101, notification); /// se lanza la
        notificación

        Intent i=new
        Intent(Alertas.this,AlertaActivada.class);

        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);//optional

        i.addFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);//will cause it to come to
        foreground

        i.putExtra("resultado1",
        resultado1); // Enviando datos a la siguiente actividad de donde fue
        presionado el boton

        startActivity(i);

        texto2.setText("¡¡El
        tiempo actual es de "+resultado1.toString()+" minutos!!");

        finish();

    }

} catch (Exception e) {
    texto2.setText(e.getMessage());
}

```

```

                                estadoalerta--; // llega a 0
despues de 4 hrs de estar checando cada 15 minutos

                                }// del while
                                }

                                private void moveTaskToBack(boolean b)
{
                                // TODO Auto-generated method
stub

                                }
                                }, MILISEGUNDOS_ESPERA);

                                }//final del boton 120 de puente Ysleta
                                }//final del onclick

}

```


Anexo 3.5

Base de datos

```

USE [clientes]
GO
/***** Object: Table [dbo].[Usuarios]      Script Date: 05/02/2013
11:54:22 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Usuarios] (
    [Id_casa] [int] IDENTITY(1,1) NOT NULL,
    [NickName] [varchar](15) NULL,
    [Contrasena] [varchar](15) NULL,
    [nivel] [varchar](2) NULL,
    [casadecambio] [varchar](15) NULL,
    [compra] [varchar](5) NULL,
    [venta] [varchar](5) NULL
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
SET IDENTITY_INSERT [dbo].[Usuarios] ON
INSERT [dbo].[Usuarios] ([Id_casa], [NickName], [Contrasena], [nivel],
[casadecambio], [compra], [venta]) VALUES (7, N'juan', N'2424', N'1',
N'Valor', N'12.00', N'12.40')
INSERT [dbo].[Usuarios] ([Id_casa], [NickName], [Contrasena], [nivel],
[casadecambio], [compra], [venta]) VALUES (8, N'manuel', N'2525',
N'1', N'Mr. Dolar', N'12.10', N'12.35')
INSERT [dbo].[Usuarios] ([Id_casa], [NickName], [Contrasena], [nivel],
[casadecambio], [compra], [venta]) VALUES (9, N'rodriguez', N'2626',
N'1', N'Patmos', N'12.07', N'12.55')
SET IDENTITY_INSERT [dbo].[Usuarios] OFF
/***** Object: Table [dbo].[estadisticasysleta]      Script Date:
05/02/2013 11:54:22 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[estadisticasysleta] (
    [dia] [varchar](15) NULL,
    [sieteam] [varchar](15) NULL,
    [nueveam] [varchar](15) NULL,
    [onceam] [varchar](15) NULL,
    [unapm] [varchar](15) NULL,
    [trespm] [varchar](15) NULL,
    [cincopm] [varchar](15) NULL,
    [sietepm] [varchar](15) NULL,
    [nuevepm] [varchar](15) NULL,
    [tiempoactual] [varchar](15) NULL,
    [tiempoactualcompleto] [varchar](15) NULL,
    [horadecaptura] [varchar](15) NULL,
    [capturahoy] [varchar](15) NULL
) ON [PRIMARY]
GO

```

```

SET ANSI_PADDING OFF
GO
INSERT [dbo].[estadisticasysleta] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'hoy', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, N'35',
N'35 min', N'0', NULL)
INSERT [dbo].[estadisticasysleta] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Monday', N'25 min', N'20 min', N'34 min', N'45 min', N'33
min', N'28 min', N'22 min', N'2 min', NULL, N'20 min', N'21:22:05
p.m.', N'11:54:13 a.m.')
INSERT [dbo].[estadisticasysleta] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Tuesday', N'5 min', N'25 min', N'20 min', N'20 min', N'8
min', N'20 min', N'25 min', N'5 min', NULL, N'40 min', N'21:22:00
p.m.', N'23:56:57 p.m.')
INSERT [dbo].[estadisticasysleta] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Wednesday', N'10 min', N'30 min', N'40 min', N'49 min', N'40
min', N'30 min', N'40 min', N'2 min', NULL, N'20 min', N'21:21:59
p.m.', N'23:56:59 p.m.')
INSERT [dbo].[estadisticasysleta] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Thursday', N'10 min', N'20 min', N'35 min', N'45 min', N'15
min', N'13 min', N'20 min', N'25 min', NULL, N'35 min', N'11:21:58
a.m.', N'11:52:00 a.m.')
INSERT [dbo].[estadisticasysleta] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Friday', N'10 min', N'16 min', N'40 min', N'45 min', N'45
min', N'25 min', N'10 min', N'25 min', NULL, N'35 min', N'21:21:48
p.m.', N'23:56:54 p.m.')
INSERT [dbo].[estadisticasysleta] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Saturday', N'25 min', N'40 min', N'45 min', N'45 min', N'45
min', N'45 min', N'45 min', N'45 min', NULL, N'45 min', N'21:21:50
p.m.', N'23:56:50 p.m.')
INSERT [dbo].[estadisticasysleta] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Sunday', N'5 min', N'30 min', N'45 min', N'45 min', N'45
min', N'40 min', N'45 min', N'30 min', NULL, N'15 min', N'21:21:51
p.m.', N'23:56:54 p.m.')
/***** Object: Table [dbo].[estadisticasisidro]      Script Date:
05/02/2013 11:54:22 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[estadisticasisidro] (
    [dia] [varchar](15) NULL,
    [sieteam] [varchar](15) NULL,
    [nueveam] [varchar](15) NULL,

```

```

        [onceam] [varchar](15) NULL,
        [unapm] [varchar](15) NULL,
        [trespm] [varchar](15) NULL,
        [cincopm] [varchar](15) NULL,
        [sietepm] [varchar](15) NULL,
        [nuevepm] [varchar](15) NULL,
        [tiempoactual] [varchar](15) NULL,
        [tiempoactualcompleto] [varchar](15) NULL,
        [horadecaptura] [varchar](15) NULL,
        [capturahoy] [varchar](15) NULL
    ) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[estadisticasisidro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Monday', N'2 hrs 42 min', N'2 hrs 40 min', N'2 hrs', N'1 hrs
40 min', N'1 hrs 40 min', N'1 hrs 50 min', N'1 hrs 50 min', N'1 hrs 40
min', NULL, N'1 hrs', N'21:21:57 p.m.', N'23:56:54 p.m.')
INSERT [dbo].[estadisticasisidro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Tuesday', N'1 hrs 40 min', N'1 hrs 55 min', N'1 hrs 55 min',
N'1 hrs 55 min', N'1 hrs 55 min', N'1 hrs 55 min', N'45 min', N'45
min', NULL, N'35 min', N'21:21:55 p.m.', N'23:56:55 p.m.')
INSERT [dbo].[estadisticasisidro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Wednesday', N'2 hrs', N'2 hrs 30 min', N'3 hrs', N'3 hrs 20
min', N'3 hrs 20 min', N'3 hrs', N'1 hrs 40 min', N'50 min', NULL,
N'20 min', N'21:21:57 p.m.', N'23:56:57 p.m.')
INSERT [dbo].[estadisticasisidro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Thursday', N'1 hrs 40 min', N'2 hrs 5 min', N'2 hrs 30 min',
N'1 hrs 40 min', N'1 hrs 15 min', N'1 hrs', N'40 min', N'30 min',
NULL, N'2 hrs 30 min', N'11:21:57 a.m.', N'11:51:57 a.m.')
INSERT [dbo].[estadisticasisidro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Friday', N'1 hrs 30 min', N'2 hrs', N'2 hrs 30 min', N'2 hrs
30 min', N'2 hrs 30 min', N'2 hrs 30 min', N'2 hrs 20 min', N'1 hrs 10
min', NULL, N'45 min', N'21:21:46 p.m.', N'23:56:50 p.m.')
INSERT [dbo].[estadisticasisidro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Saturday', N'50 min', N'1 hrs 10 min', N'1 hrs 30 min', N'1
hrs 15 min', N'1 hrs 30 min', N'1 hrs 45 min', N'1 hrs 40 min', N'1
hrs 50 min', NULL, N'1 hrs 45 min', N'21:21:48 p.m.', N'23:56:48
p.m.')
INSERT [dbo].[estadisticasisidro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Sunday', N'45 min', N'50 min', N'1 hrs', N'1 hrs 50 min',
N'2 hrs 30 min', N'3 hrs', N'3 hrs 15 min', N'3 hrs', NULL, N'3 hrs 5
min', N'21:21:50 p.m.', N'23:56:53 p.m.')
INSERT [dbo].[estadisticasisidro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])

```

```

VALUES (N'hoy', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
N'150', N'2 hrs 30 min', N'0', N'')
/***** Object: Table [dbo].[estadisticascentro]      Script Date:
05/02/2013 11:54:22 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[estadisticascentro] (
    [dia] [varchar](15) NULL,
    [sieteam] [varchar](15) NULL,
    [nueveam] [varchar](15) NULL,
    [onceam] [varchar](15) NULL,
    [unapm] [varchar](15) NULL,
    [trespm] [varchar](15) NULL,
    [cincopm] [varchar](15) NULL,
    [sietepm] [varchar](15) NULL,
    [nuevepm] [varchar](15) NULL,
    [tiempoactual] [varchar](15) NULL,
    [tiempoactualcompleto] [varchar](15) NULL,
    [horadecaptura] [varchar](15) NULL,
    [capturahoy] [varchar](15) NULL
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[estadisticascentro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'hoy', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, N'29',
N'29 min', N'0', NULL)
INSERT [dbo].[estadisticascentro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Monday', N'10 min', N'15 min', N'25 min', N'40 min', N'48
min', N'48 min', N'5 min', N'5 min', NULL, N'20 min', N'21:22:03
p.m.', N'11:54:13 a.m.')
INSERT [dbo].[estadisticascentro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Tuesday', N'5 min', N'25 min', N'20 min', N'14 min', N'5
min', N'5 min', N'5 min', N'3 min', NULL, N'10 min', N'21:21:59 p.m.',
N'23:56:57 p.m.')
INSERT [dbo].[estadisticascentro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Wednesday', N'5 min', N'30 min', N'35 min', N'45 min', N'29
min', N'13 min', N'5 min', N'5 min', NULL, N'5 min', N'21:21:58 p.m.',
N'23:56:58 p.m.')
INSERT [dbo].[estadisticascentro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Thursday', N'10 min', N'25 min', N'29 min', N'32 min', N'33
min', N'25 min', N'26 min', N'26 min', NULL, N'29 min', N'11:21:58
a.m.', N'11:51:59 a.m.')
INSERT [dbo].[estadisticascentro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Friday', N'5 min', N'25 min', N'30 min', N'42 min', N'48

```

```

min', N'49 min', N'38 min', N'15 min', NULL, N'20 min', N'21:21:48
p.m.', N'23:56:53 p.m.')
INSERT [dbo].[estadisticascentro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Saturday', N'24 min', N'41 min', N'41 min', N'25 min', N'10
min', N'30 min', N'40 min', N'30 min', NULL, N'40 min', N'21:21:49
p.m.', N'23:56:49 p.m.')
INSERT [dbo].[estadisticascentro] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Sunday', N'5 min', N'25 min', N'30 min', N'44 min', N'20
min', N'25 min', N'30 min', N'40 min', NULL, N'15 min', N'21:21:51
p.m.', N'23:56:54 p.m.')
/***** Object: Table [dbo].[estadisticasamericas]      Script Date:
05/02/2013 11:54:22 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[estadisticasamericas] (
    [dia] [varchar](15) NULL,
    [sieteam] [varchar](15) NULL,
    [nueveam] [varchar](15) NULL,
    [onceam] [varchar](15) NULL,
    [unapm] [varchar](15) NULL,
    [trespm] [varchar](15) NULL,
    [cincopm] [varchar](15) NULL,
    [sietepm] [varchar](15) NULL,
    [nuevepm] [varchar](15) NULL,
    [tiempoactual] [varchar](15) NULL,
    [tiempoactualcompleto] [varchar](15) NULL,
    [horadecaptura] [varchar](15) NULL,
    [capturahoy] [varchar](15) NULL
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
INSERT [dbo].[estadisticasamericas] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Monday', N'45 min', N'34 min', N'40 min', N'40 min', N'25
min', N'5 min', N'5 min', N'5 min', NULL, N'23 min', N'21:22:00 p.m.',
N'11:54:12 a.m.')
INSERT [dbo].[estadisticasamericas] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Tuesday', N'15 min', N'23 min', N'23 min', N'33 min', N'15
min', N'10 min', N'10 min', N'10 min', NULL, N'24 min', N'21:21:57
p.m.', N'23:56:56 p.m.')
INSERT [dbo].[estadisticasamericas] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Wednesday', N'23 min', N'33 min', N'42 min', N'42 min', N'33
min', N'35 min', N'15 min', N'10 min', NULL, N'31 min', N'21:21:58
p.m.', N'23:56:58 p.m.')
INSERT [dbo].[estadisticasamericas] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])

```

```

VALUES (N'Thursday', N'19 min', N'27 min', N'42 min', N'40 min', N'42
min', N'42 min', N'38 min', N'35 min', NULL, N'42 min', N'11:21:58
a.m.', N'11:51:58 a.m.')
INSERT [dbo].[estadisticasamericas] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Friday', N'25 min', N'25 min', N'44 min', N'44 min', N'44
min', N'46 min', N'41 min', N'41 min', NULL, N'45 min', N'21:21:47
p.m.', N'23:56:51 p.m.')
INSERT [dbo].[estadisticasamericas] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Saturday', N'25 min', N'35 min', N'35 min', N'43 min', N'29
min', N'35 min', N'35 min', N'38 min', NULL, N'45 min', N'21:21:49
p.m.', N'23:56:49 p.m.')
INSERT [dbo].[estadisticasamericas] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'Sunday', N'5 min', N'30 min', N'30 min', N'40 min', N'42
min', N'15 min', N'18 min', N'25 min', NULL, N'45 min', N'21:21:50
p.m.', N'23:56:53 p.m.')
INSERT [dbo].[estadisticasamericas] ([dia], [sieteam], [nueveam],
[onceam], [unapm], [trespm], [cincopm], [sietepm], [nuevepm],
[tiempoactual], [tiempoactualcompleto], [horadecaptura], [capturahoy])
VALUES (N'hoy', NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, N'42',
N'42 min', N'0', NULL)
/***** Object: Default [DF_estadisticasamericas_horadecaptura]
Script Date: 05/02/2013 11:54:22 *****/
ALTER TABLE [dbo].[estadisticasamericas] ADD CONSTRAINT
[DF_estadisticasamericas_horadecaptura] DEFAULT (getdate()) FOR
[horadecaptura]
GO

```