**Subdirección Académica**

**Departamento de Sistemas y Computación**

**Ingeniería en Sistemas Computacionales**

**Semestre: Agosto – Diciembre 2016**

**Materia: Lenguajes y Autómatas 2 (6SC7C)**

**Sistema web para búsqueda e identificación de actividades y lugares comunes de entes registrados (ITWacho)**

**Nombre del alumno:**

**Salcedo Morales José Manuel (13211419)**

**Nombre del catedrático:**

**Virgilio Rosendo Pérez Pérez**

**Fecha de entrega:**

**Tijuana, Baja California a 28 de Noviembre de 2016**

Índice

[Introducción 3](#_Toc468119015)

[Metodología de trabajo 3](#_Toc468119016)

[Planteamiento del problema 3](#_Toc468119017)

[Objetivos 4](#_Toc468119018)

[General 4](#_Toc468119019)

[Específicos 4](#_Toc468119020)

[Justificación 4](#_Toc468119021)

[Estado de arte 4](#_Toc468119022)

[Análisis de sensibilidad 5](#_Toc468119023)

[Costo 6](#_Toc468119024)

[Marco Teórico 6](#_Toc468119025)

[Identificación de las variables 6](#_Toc468119026)

[Metodología 7](#_Toc468119027)

[Análisis del problema 7](#_Toc468119028)

[Diseño del algoritmo 7](#_Toc468119029)

[Principal 7](#_Toc468119030)

[Base de datos 7](#_Toc468119031)

[BackEnd 8](#_Toc468119032)

[Depuración del procedimiento. 9](#_Toc468119033)

[Programación aplicada 9](#_Toc468119034)

[Desarrollo 10](#_Toc468119035)

[FrontEnd 10](#_Toc468119036)

[BackEnd 13](#_Toc468119037)

[Base de datos 15](#_Toc468119038)

[Implementación. Optimización 15](#_Toc468119039)

[Reducción de archivos 15](#_Toc468119040)

[Llamadas en paralelo (No comprobado) 19](#_Toc468119041)

[Conclusión y trabajo futuro 20](#_Toc468119042)

[Bibliografía 20](#_Toc468119043)

# Introducción

Con el avance de la tecnología, lo más probable que cualquier persona piense, es que todos los procesos cada vez son más autónomos y sencillos de realizar, pero la tecnología nunca puede reemplazar el hecho de que las personas se ocupan comunicar físicamente, para hablar de temas un poco más importantes. Y ese es el objetivo de la aplicación a desarrollar, dar a entender que nada puede cambiar el hecho de que es importante comunicarse físicamente, para así no obtener ningún malentendido. Con ello, se realizara un sistema web donde los usuarios que estén dentro del sistema, puedan pedir ayuda para localizar a un ente, objeto, etcétera, esto no siendo geolocalización, sino trabajando con la información que otras personas proporcionen.

## Metodología de trabajo

Para la metodología a implementar, se pretende hacer de uso del prototipado.  
Con el hecho de que se puede dar una versión mínima a presentar, y gradualmente dar a conocer mejores versiones de ello. Esto siendo muy conveniente, ya que el llenado de información se da de poco en poco.

# Planteamiento del problema

Siendo estudiantes, trabajadores, o cualquier tipo de persona con una actividad constante, que te obliga a no estar en una ubicación fija, a veces resulta difícil para otras personas conocer o saber dónde se encuentran aquellas personas con las que ocupan hablar físicamente, por lo cual debido a que no hay aplicaciones populares que resuelvan este problema, hay necesidad de un sistema para conocer la ubicación de las personas de interés propio.

# Objetivos

## General

Se desarrollara e implementar un sistema web para que las personas puedan investigar información previa sobre donde se localiza una persona, y si no se encuentra de la primera forma, tener indicios para encontrarlo con facilidad. Aplicado a una Institucion Educativa.

## Específicos

* Creación de aplicación web usando un formato adaptable apoyándose de HTML/css y JavaScript además de algunas librerías como Jquery y/o bootstrap.
* Diseño, creación, modificación y montaje de la base de datos.
* Configuración y montaje de un BackEnd con un lenguaje de programación apropiada para lo que se desea desarrollar.
* Configuración de servidor web para patrocinar el sistema a otros usuarios.

# Justificación

Dentro del Instituto Tecnológico de Tijuana, en ocasiones se requiere encontrar a una persona, ya sea por motivos académicos y/o personales. Sin embargo, no siempre está disponible en el lugar que se querría encontrar.

Por ello, si alguien más puede saber de su localización, otras personas pueden saber de sus previos lugares de encuentro. Y si están grabadas digitalmente, mucho mejor.

# Estado de arte

En cuanto a aplicaciones web, las tendencias actuales se resumen algunas, gracias a 40DeFiebre[[1]](#footnote-1):

* Nuevo Diseño Flat: "Google nos lo presentó hace ya (casi) un par de años, la digievolución de lo que ya era una realidad. Lo llamó Material design. Dicen que para su creación se inspiraron en la tinta y el papel.
* Tipografía: "Otra de las tendencias en diseño web que veremos cada vez más frecuentemente es el uso expresivo de la tipografía. Se utilizará como forma y no como medio para comunicar, o dicho de otra forma, se ilustrará con letras."
* Patrones UI: "Estos patrones guían al usuario mientras navega por tu web. Te solucionarán más de una papeleta. Úsalos, funcionan."

En cuanto a las tendencias a utilizar, Python es una de las cuales se apoyan más recientemente[[2]](#footnote-2):

"El desarrollo de software moderno con soluciones como el cuarteto MEAN (MongoDB, ExpressJS, AngularJS y Node.js) o el uso de tecnologías de Big Data como Spark impulsan otro tipo de lenguajes de programación como JavaScript, Python o Scala por encima de Java, una sintaxis que hace años era una puerta de entrada casi segura en el mercado profesional del desarrollo. Hoy todavía sigue siéndolo, pero su poderío mengua con el paso del tiempo.

Una simple búsqueda en indeed, el portal de empleo y tendencias, marca curvas claras para todas las sintaxis que hemos mencionado: Java claramente a la baja, JavaScript y sobre todo Python y Scala, al alza. Lógicamente, el mercado americano es muy distinto al mercado español. Eso es importante."

Por último, también una tecnología a implementar son los lenguajes de programación del lado del cliente. JavaScript liderando la marcha[[3]](#footnote-3):

"JavaScript es probablemente el “nuevo BASIC”. Así como en los años ochenta del siglo pasado, cuando empezaban a surgir las computadoras de 8 bits que corrían casi todas algún tipo de intérprete de BASIC, hoy JavaScript es el equivalente en la tecnología moderna para crear páginas web, escribir juegos relativamente simples y es sin duda, el lenguaje de scripts más popular de la web. Los navegadores más populares tienen incluido un intérprete de JavaScript y probablemente una inmensa mayoría de sitios web usen algún elemento de JavaScript en sus páginas. Ahora, además, se ha incorporado Node.js, una tecnología de servidores que permite a JavaScript comunicación en tiempo real."

# Análisis de sensibilidad

En cuanto al tiempo, se da a consideración que se invertirán mínimo 3 horas de desarrollo por semana por cada persona involucrada, en un promedio de inversión de 10 horas por semana.  
Lo anterior con un tiempo de competición de 5 meses.

# Costo

Debido al proyecto será aplicado a una institución educativa (y para fines de investigación), se dará el cobro de $15 dólares por hora, esto siendo considerado como un apoyo a los desarrolladores.

Con el costo y el tiempo a invertir, se puede hacer los cálculos:

10 horas por semana x 5 meses = 200 horas para terminar. 200 horas x $15 dólares por hora = $3000 dólares por desarrollo completo.

Los gastos incluyendo el mantenimiento del proyecto siendo:

|  |  |
| --- | --- |
| Actividad | Costo (dólares) |
| Desarrollo | $3000.00 |
| Servidor | ~$500.00 |
| Soporte Técnico | Variado (No se cubre) |
| Recursos (Electricidad, componentes de reemplazo, etcétera) | Variado (No se cubre) |

# Marco Teórico

Se pretende desarrollar un software que pueda ser aplicado en una zona limitada geográficamente.  
Esto pudiendo ser:

* Una institución.
* Un edificio.
* Departamentos.

Con lo anterior, será de uso en una institución educativa, con el propósito de apoyar en la localización de personas que posiblemente no se encuentren en su lugar de estadía común.

## Identificación de las variables

Variable dependiente: Ubicación de las personas interesadas a encontrar y sus actividades que realizan con más frecuencia. Variable independiente: El tiempo que tardaría en llegar en el lugar donde se encuentra la persona a buscar. Variables de control: Todos los partícipes en una zona. En este caso en una institución educativa.

# Metodología

## Análisis del problema

1. Formular.
2. Se necesita un sistema el cual proporcione información sobre los lugares y actividades comunes de las personas que tramitan un lugar específico. Para este sistema una institución educativa.
3. Resultados esperados.
4. Una plataforma donde:
   1. Se pueda dar de alta información donde un usuario pueda dar de alta los lugares y actividades comunes de una persona.
   2. Revisar la información previa que se ha dado de alta.
5. Datos disponibles.
6. Personas.
7. Lugares.
8. Actividades.
9. Restricciones.
10. No repetir las personas que existen. Evitar redundancia.
11. Las actividades y lugares que realicen las personas solo deben ser las que sean dentro del lugar.
12. Procesos necesarios.
13. Alta y consulta de:
    1. Personas.
    2. Actividades.
    3. Lugares.

## Diseño del algoritmo

### Principal

1. Inicio.
2. Página de inicio.
3. Personas <- [].
4. Vacíen, "Personas".
5. Desplegar, Personas.
6. Localizaciones <- [], Actividades <- [].
7. Desplegar "Decida persona", Persona <- "".
8. Desplegar "Desea actividades?", Actividad <- bool
9. Desplegar "Desea localizaciones?", Localización <- bool
10. BackEnd, "Información".
11. Desplegar Localizaciones, Actividades.
12. Si salir = true, paso 13. Sino, paso
13. Fin.

### Base de datos

#### Personas

1. Inicio.
2. Personas <- [].
3. Extraer, Personas.
4. Retornar, Personas.
5. Fin.

#### Información

1. Inicio.
2. Persona <- "", Localización <- bool, Actividad <- bool.
3. Localizaciones <- [], Actividades <- [].
4. Extraer, Localizaciones, Actividades.
5. Si Localización = true && Actividad = true, paso
6. Sino, paso 8.
7. Regresar Localizaciones, Actividades.
8. Paso 14.
9. Si Localización = true && Actividad = false, paso
10. Sino, paso 11.
11. Regresar Localizaciones.
12. Paso 14.
13. Si Localización = false && Actividad = true, paso 12. Sino, paso 14.
14. Regresar Actividades.
15. Paso 14.
16. Fin.

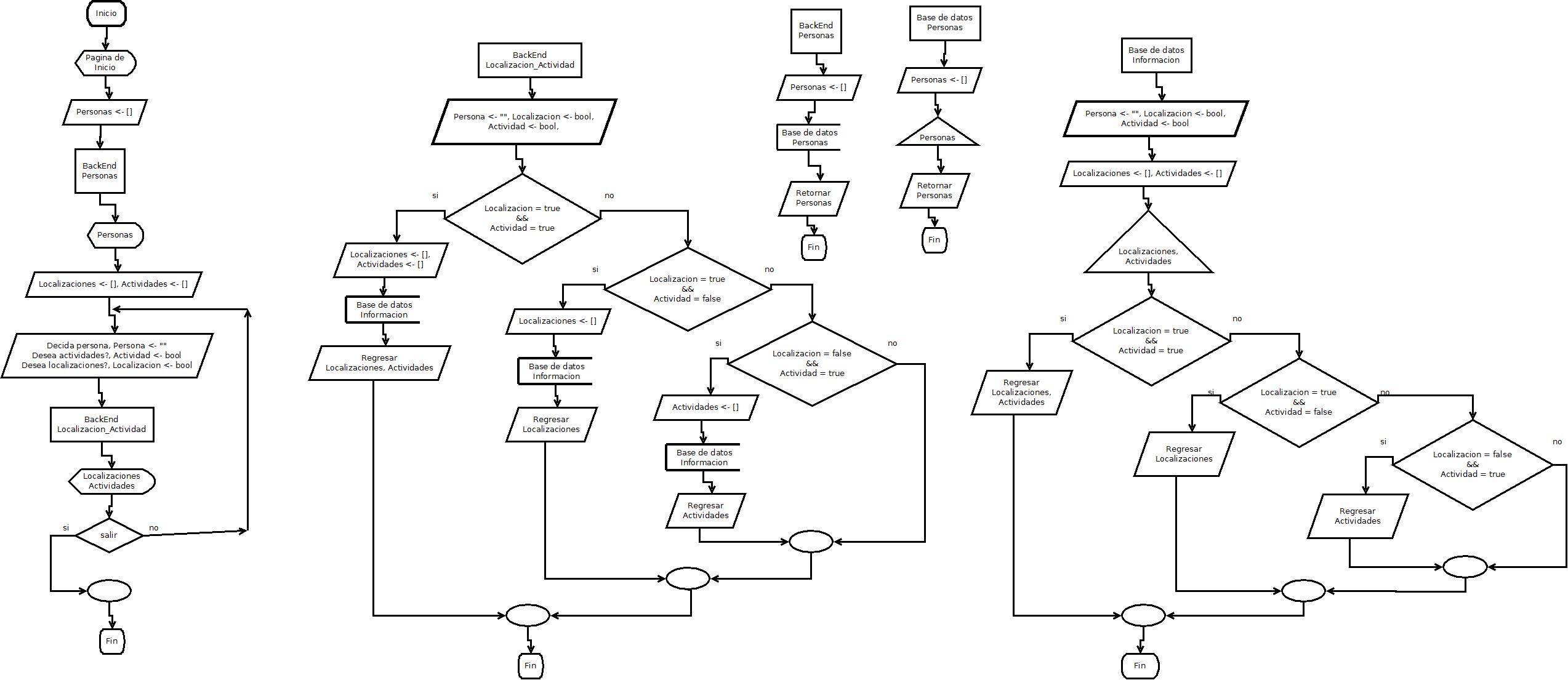
### BackEnd

#### Personas

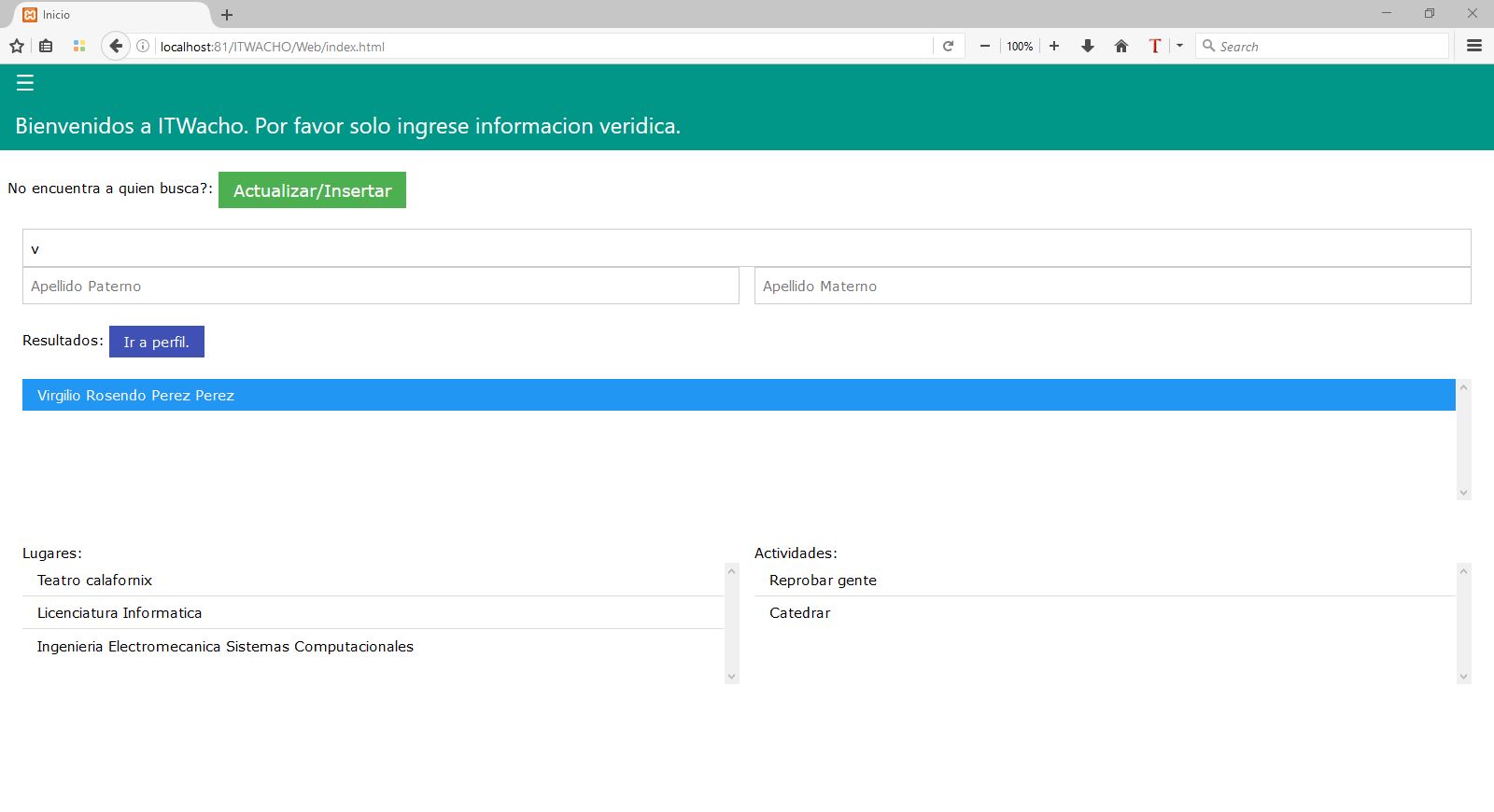
1. Inicio.
2. Personas <- [].
3. Base de datos, Personas.
4. Retornar, Personas.
5. Fin.

#### Localizacion\_Actividad

1. Inicio.
2. Persona <- "", Localización <- bool, Actividad <- bool.
3. Localización = true && Actividad = true, paso 4. Sino, paso 7.
4. Localizaciones <- [], Actividades <- [].
5. Base de datos, Información.
6. Regresar Localizaciones, Actividades.
7. Localización = true && Actividad = false, paso 8. Sino, paso 11.
8. Localizaciones <- [].
9. Base de datos, Información.
10. Regresar Localizaciones.
11. Localización = false && Actividad = true, paso 12. Sino, paso 15.

**Diagrama de flujo** 

## Depuración del procedimiento.

Debajo una captura del funcionamiento de la primera versión de la pantalla de inicio. Aunque faltan las funciones de poder pedir la búsqueda de un ente y la vista de perfil. 

# Programación aplicada

**Programación Imperativa**  
**Programación Orientada a Objetos**  
Considerando que se van a introducir y leer datos ingresados, y los datos siendo ingresados de acuerdo a opiniones de usuarios, se realiza como una programación imperativa. Los lenguajes de programación principal siendo JavaScript y PHP. Incluyendo un lenguaje de programación especifico, cualquiera que se desee. En este caso, con uso de MySql.

# Desarrollo

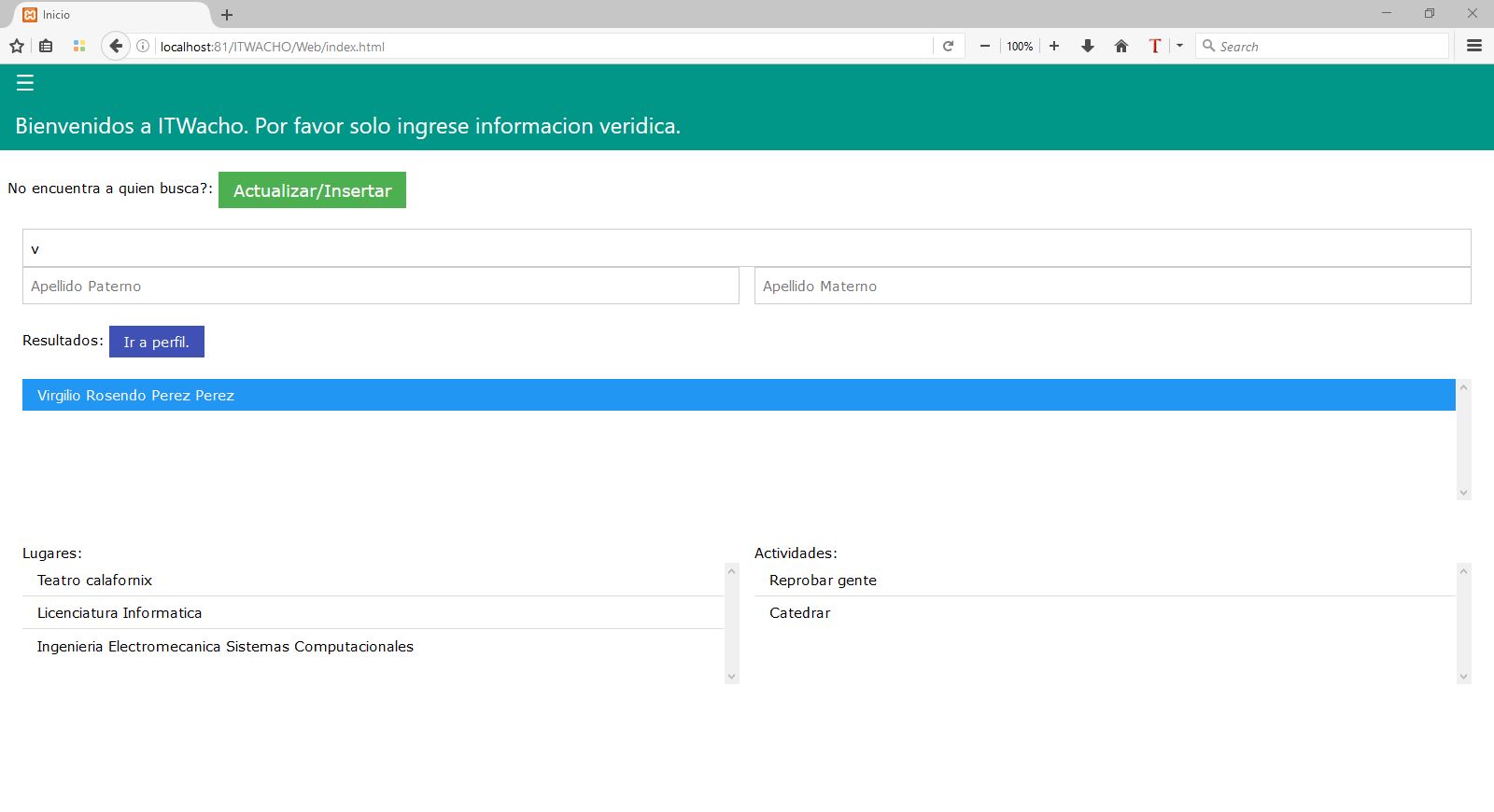
Para cumplir con los requerimientos, se da la necesidad de cubrir lo siguiente para un sistema web:

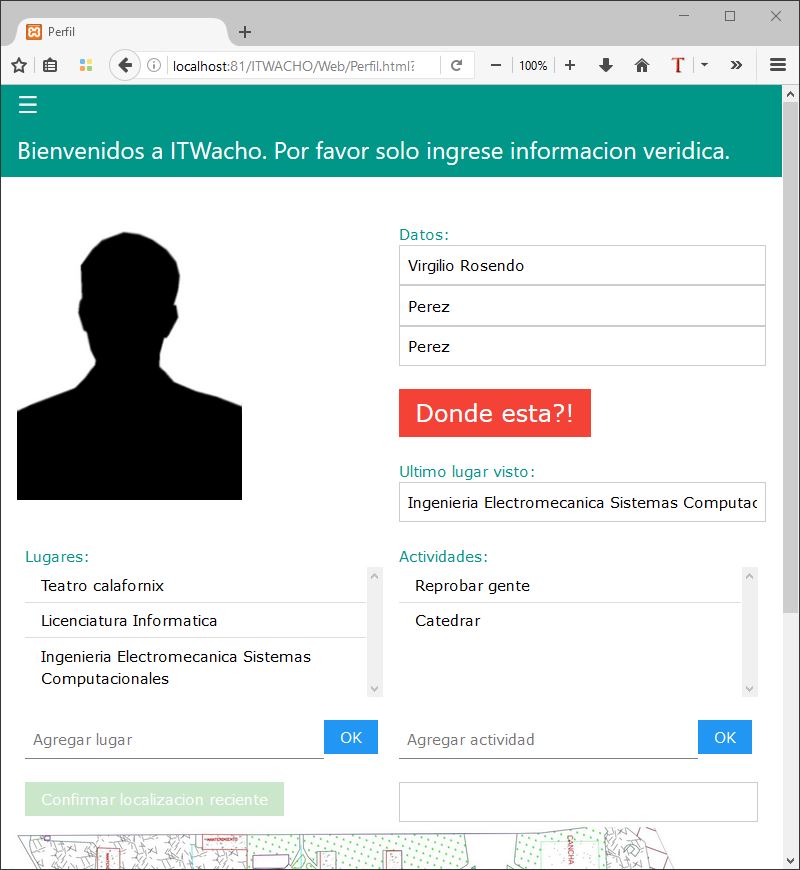
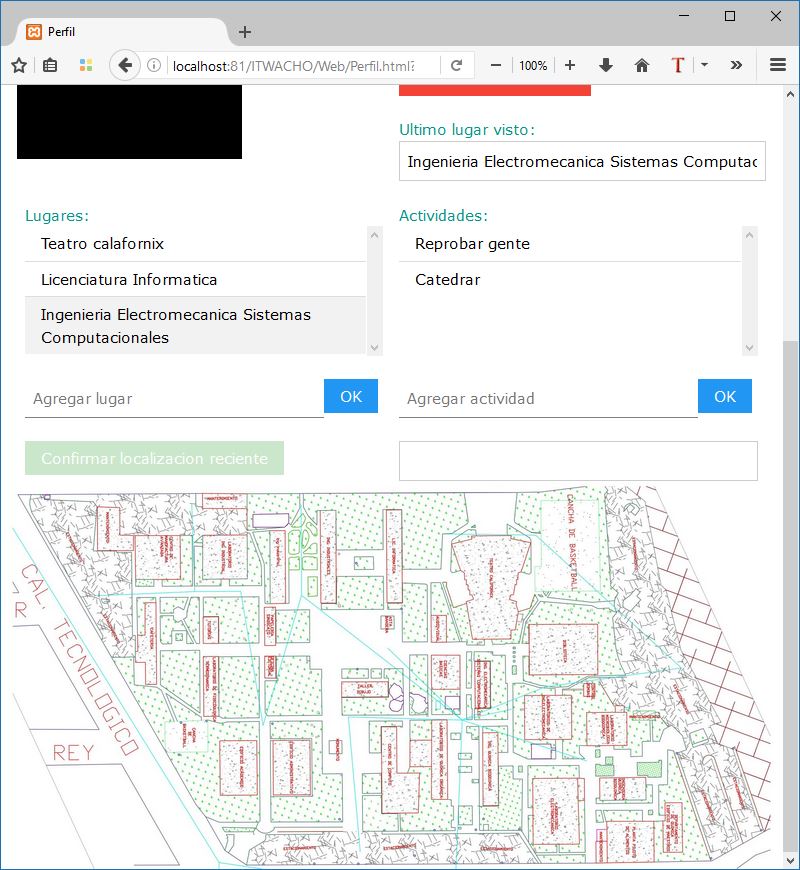
* FrontEnd
* BackEnd
* Base de datos

Para ello se proporcionan los diferentes resultados.

## FrontEnd

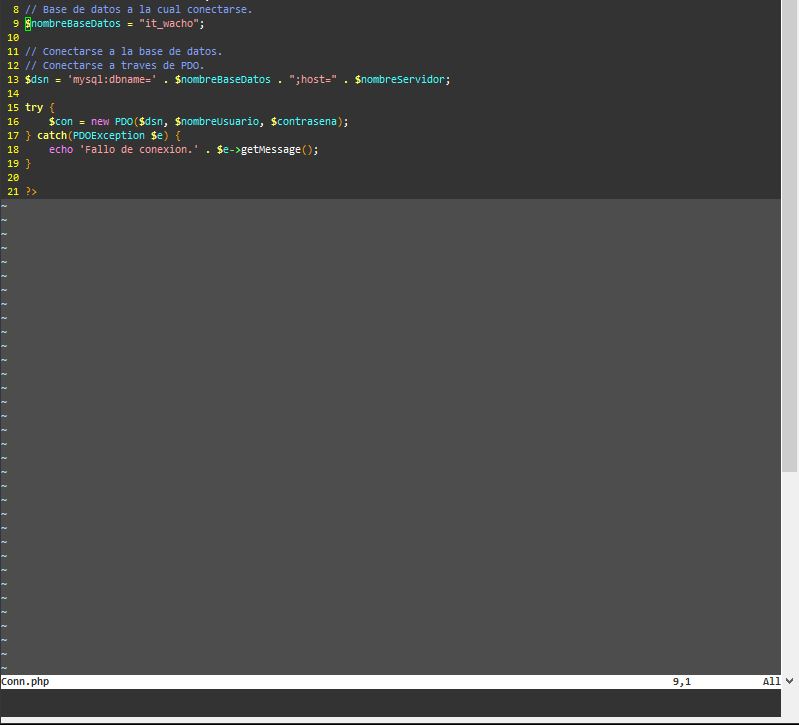
Para las vistas, se da un detalle minimalista, que da la información que se requiere sin muchas animaciones. Esto para mejorar el rendimiento de renderizado de la página hacia al usuario.

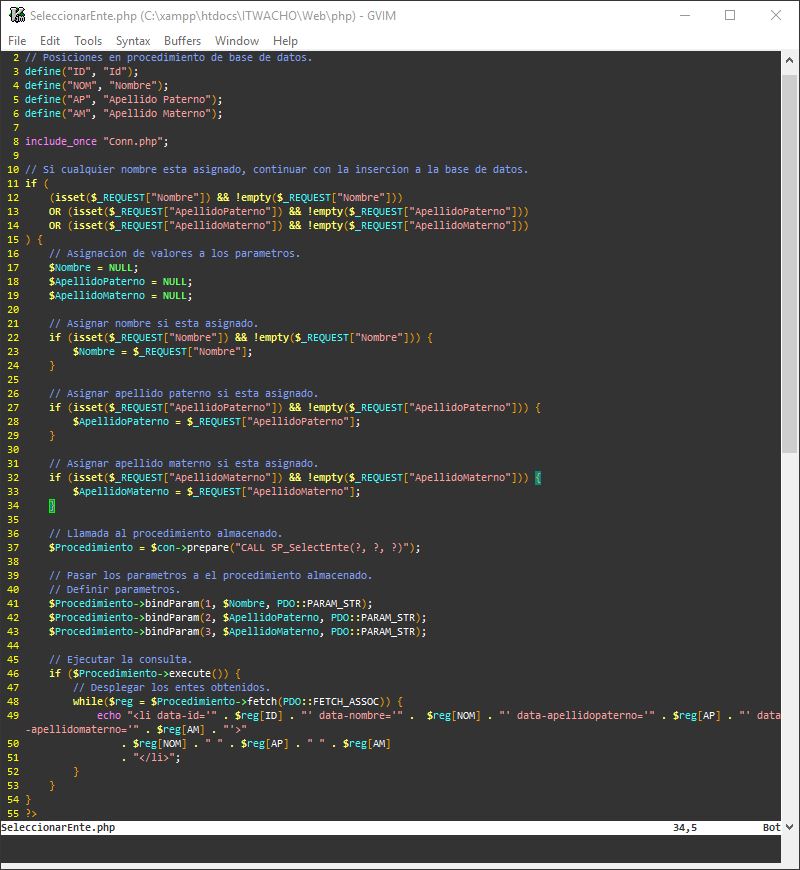
**Pantallas** 

## BackEnd

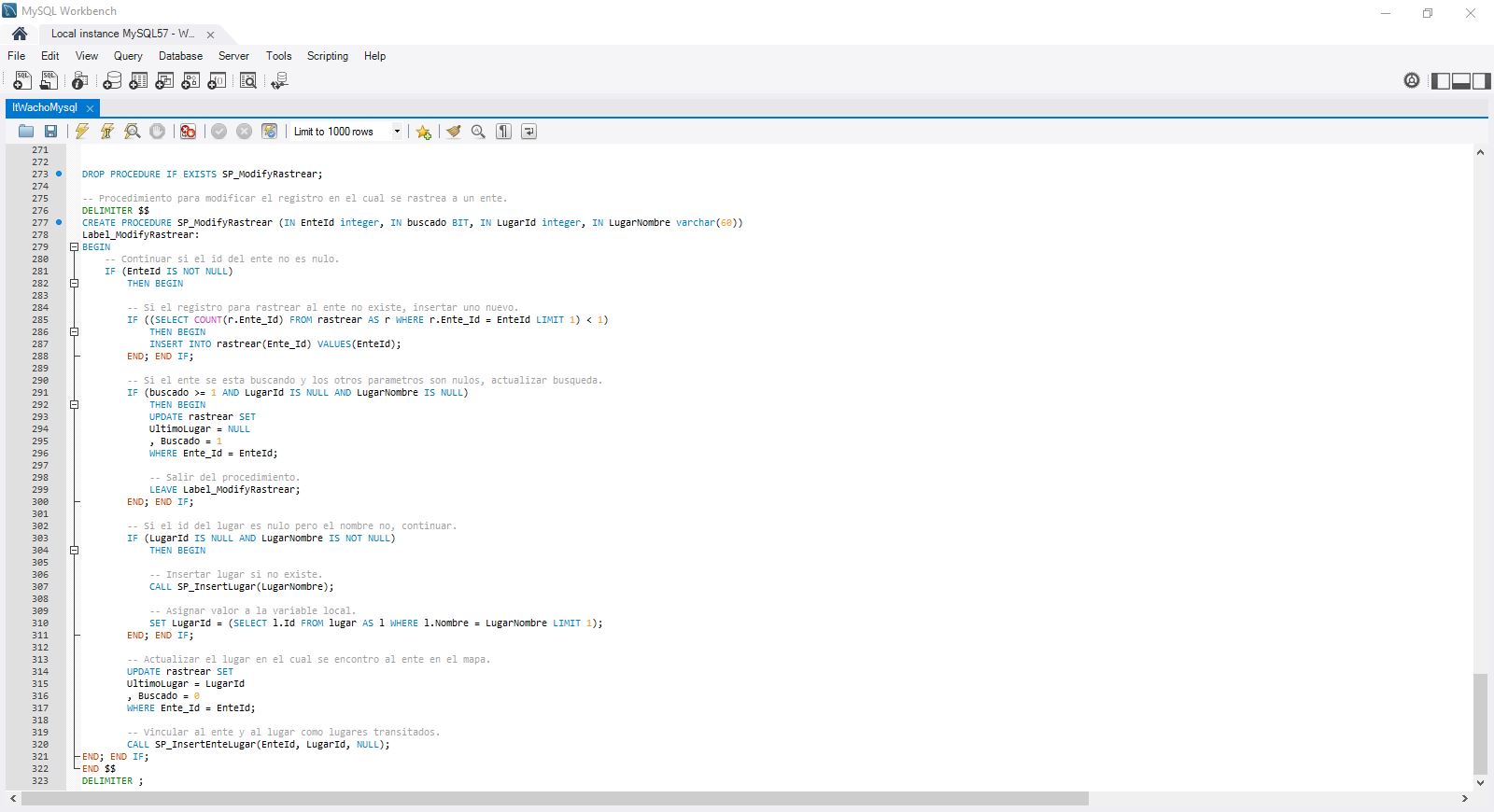
En cuanto al manejo de datos, se manejan solo conexiones y operaciones con bases de datos, con una metodología de programación orientada a objetos. Esto siendo para que, aun con el objetivo de utilizar un solo sistema manejador de bases de datos, se pueda adaptar si se desea migrar a otra plataforma, o inclusive usar otras en conjunto.  
Todas las operaciones siendo controladas por medio de procedimientos almacenados, dando otra capa de seguridad y usabilidad a la aplicación.

**Pantallas** 



## Base de datos

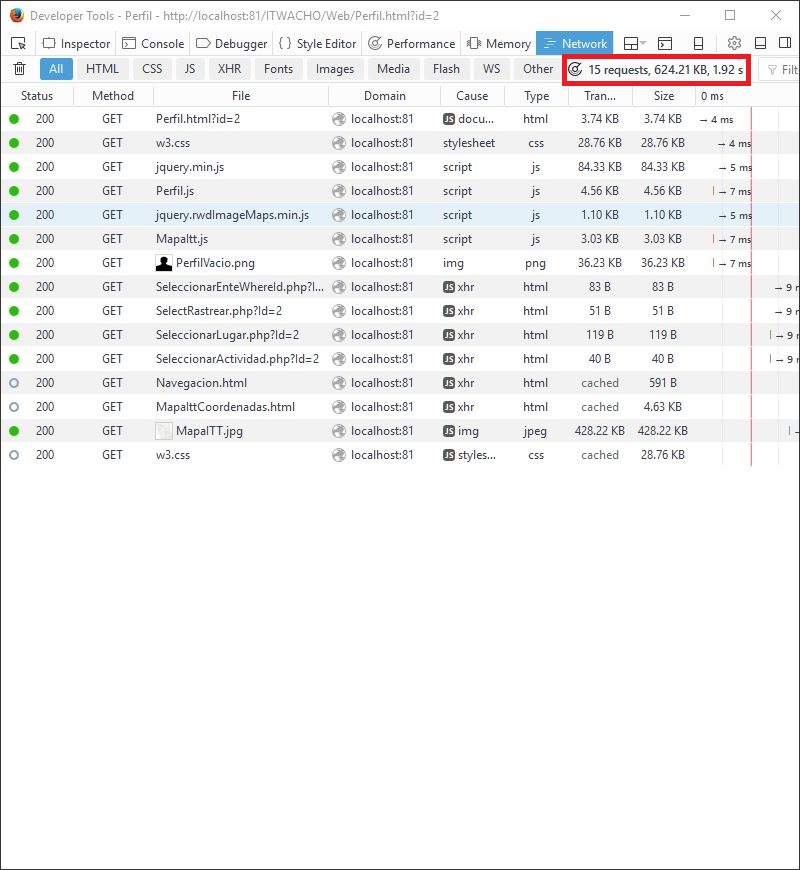
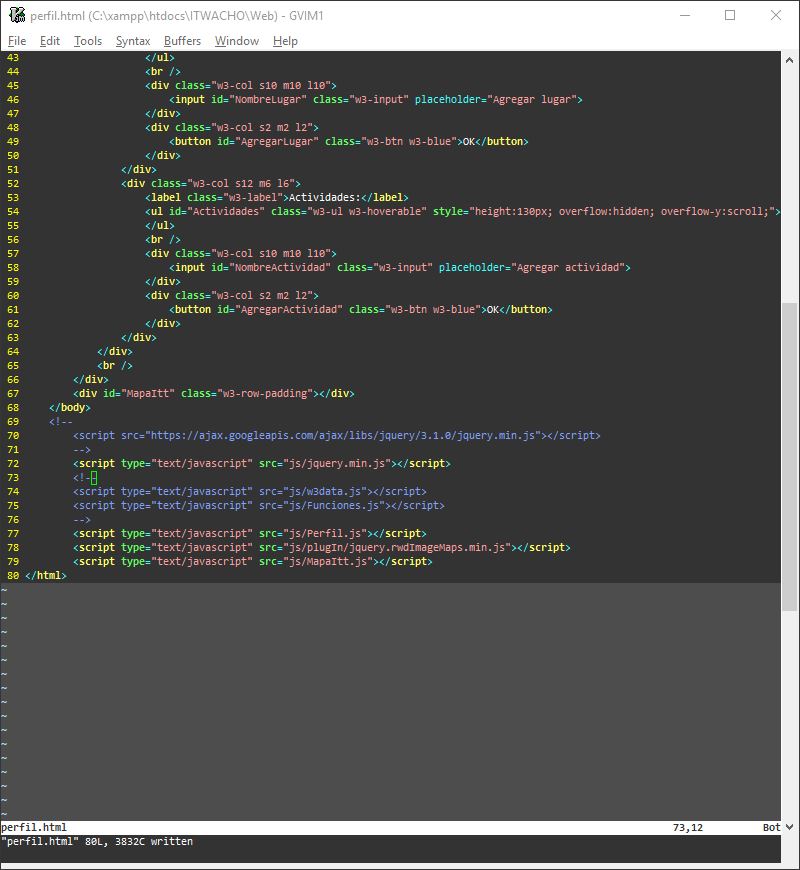
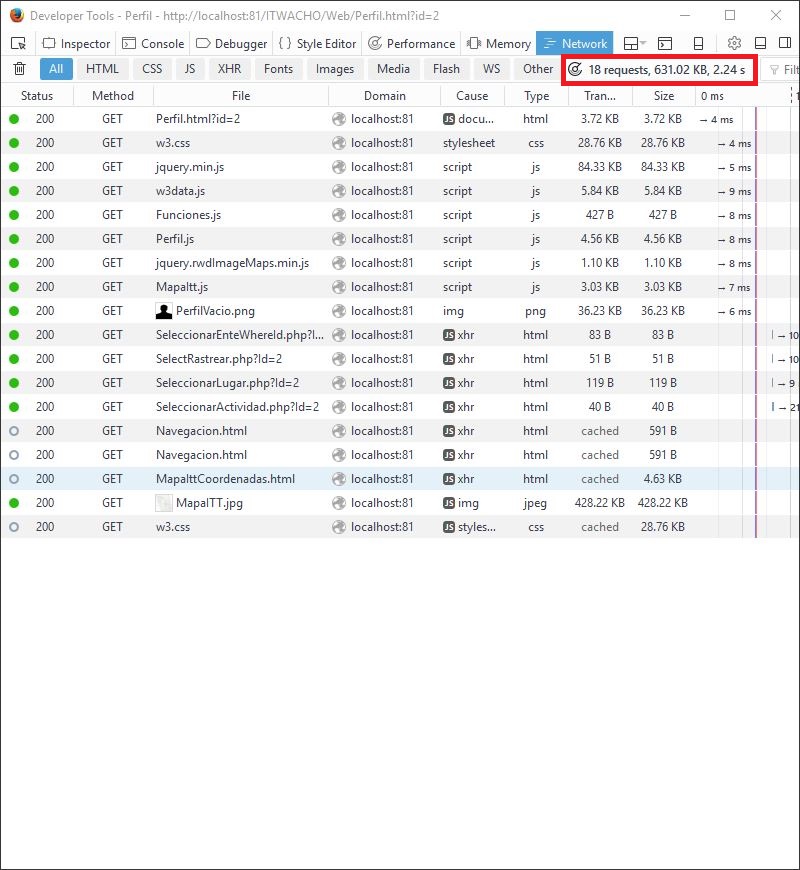
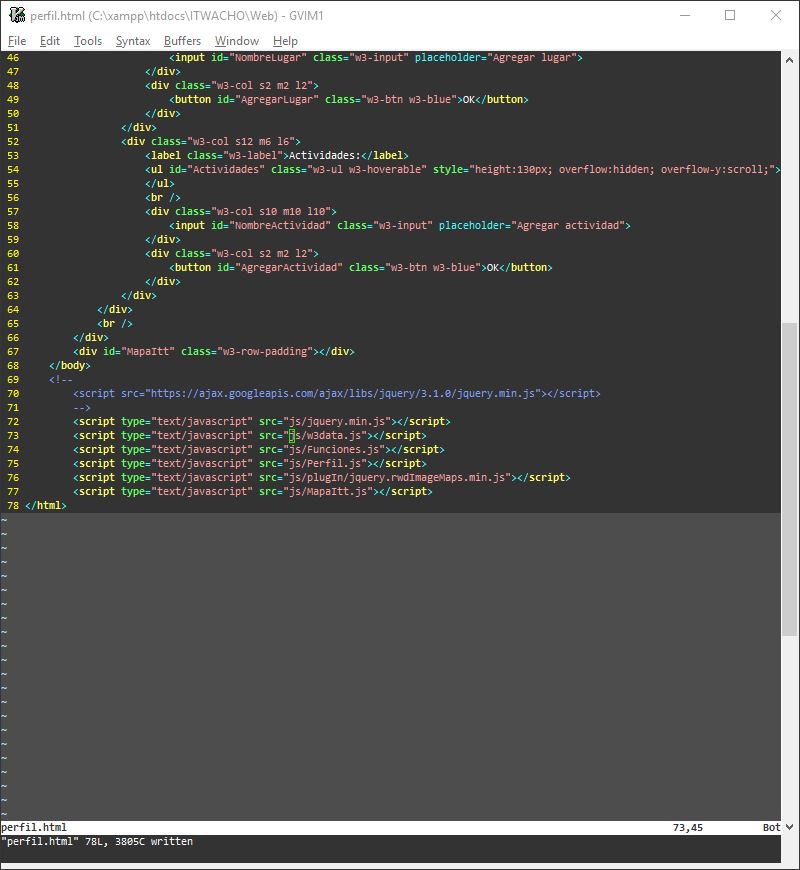
Para el manejador de la base de datos, se utiliza MySql. Utilizado este por su compatibilidad con varios sistemas operativos (Windows y variantes de Unix en específico).

**Pantalla Ejemplo** 

# Implementación. Optimización

## Reducción de archivos

Para darle funcionalidad a una página web, se implementan archivos que se ejecutan del lado del cliente. Si hay muchos archivos al cargar una página, esta se alenta. Para mejorar el rendimiento, módulos puestos en otros archivos se pueden poner en uno solo o en una cantidad menor. De esta forma, la página se renderiza mas rápido.

**Ejemplo de velocidad**    

## Llamadas en paralelo (No comprobado)

Un punto no comprobado es la optimización utilizando llamadas paralelas de archivos. Esto, utilizando un archivo de un patrocinado (CDN). Debido a la mínima cantidad de archivos en utilidad, este punto no se puede comprobar con facilidad.

# Conclusión y trabajo futuro

Aunque hay ya un gran avance para reconocer las actividades y lugares comunes en los cuales participan las personas, aun es necesario implementar algoritmos/módulos que ayuden a prevenir la redundancia de datos. O bien, dar a los usuarios los lugares y actividades verídicas de otras personas definido en cuanto a "feedback" de quienes utilicen la información.

Aun así, la idea y utilidad mínima ya está definida y usable.

# Bibliografía

1. Michelone, M. L. (2016, March 28). Los lenguajes de programación más demandados en 2016. Retrieved September 07, 2016, from https://www.unocero.com/2016/03/28/los-nueve-lenguajes-mas-demandados-en-el-2016/
2. Rodriguez, C. (n.d.). Tendencias en diseño web para 2016 (con ejemplos). Retrieved September 07, 2016, from https://www.40defiebre.com/tendencias-diseno-web-2016/
3. Tendencias en desarrollo de software para 2016. (2016, February 01). Retrieved September 06, 2016, from https://bbvaopen4u.com/es/actualidad/tendencias-en-desarrollo-de-software-para-2016
4. Capítulo 2 Marco Teórico. (n.d.). Retrieved September 20, 2016, from http://catarina.udlap.mx/u\_dl\_a/tales/documentos/lis/flores\_r\_mi/capitulo2.pdf

1. https://www.40defiebre.com/tendencias-diseno-web-2016/ [↑](#footnote-ref-1)
2. https://bbvaopen4u.com/es/actualidad/tendencias-en-desarrollo-de-software-para-2016 [↑](#footnote-ref-2)
3. https://www.unocero.com/2016/03/28/los-nueve-lenguajes-mas-demandados-en-el-2016/ [↑](#footnote-ref-3)