SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Oaxaca
DIVISIÓN DE ESTUDIOS PROFESIONALES

INSTITUTO TECNOLÓGICO DE OAXACA

División De Estudios Profesionales.

Departamento De Sistemas y Computación

TESIS

"Diseño y Aplicación de una Plataforma para la Captación, Difusión y Estudio de las preferencias, expectativas y precepciones de los turistas en Oaxaca"

Que para obtener el título de:

Ingeniero en Sistemas Computacionales

PRESENTAN:

Negrete Pérez Maricela

Vásquez Hernández Zuriel Juan

Velasco Cruz Rosario Elizabeth

ASESOR:

M.A.T.I. Claribel Benítez Quecha

Comisión revisora.

M.C.C. Anayansi Cristina Hernández Abrego

L.I. Idarh Claudio Matadamas Ortiz

DRA. Marisol Altamirano Cabrera

Oaxaca de Juárez, Oaxaca. Marzo del 2019.

Agradecimientos

El amor recibido, la dedicación y la paciencia con la que cada día se preocupaban mis padres por mi avance en la carrera, es simplemente único y se refleja en la vida de un hijo.

Gracias a mis padres por ser los principales promotores de mis sueños, gracias a ellos por cada día confiar y creer en mí, gracias a mi madre por todo su esfuerzo, su trabajo, las palabras de aliento y por su inmenso amor; gracias a mi padre por siempre desear y anhelar lo mejor para mí, por cada consejo y por cada una de sus palabras que me guiaron durante mi vida.

Gracias a Dios por la vida de mis padres, hermanos y de mi abuelita, también porque cada día bendice mi vida con la hermosa oportunidad de estar y disfrutar al lado de las personas que yo más amo en mi vida.

Gracias a mi abuelito Félix que, aunque ya no está para verme realizar este sueño, formo parte de esto que hoy estoy logrando, su fe en dios y sus oraciones me permitieron no rendirme en situaciones difíciles de la carrera.

Gracias a la vida por este nuevo triunfo, gracias a mis amigos de residencia, a quienes tuve la dicha de conocer en este camino que no iniciamos juntos pero que si culminaremos y gracias a todas las personas que me apoyaron y creyeron en la realización de este sueño.

Maricela Negrete Pérez

Agradecimientos

Agradecida con dios por no olvidarse de mí, por escuchar mis suplicas y ruegos, ante ti señor quien en algún momento escucho mis llantos, te doy gracias por iluminar mis días más oscuros y no dejar perder la fe en ti señor, gracias por permitir ver a mi familia, amigos y seres queridos con salud todos los días.

A mi familia, como un testimonio de cariño y eterno agradecimiento por mi existencia, principios y excelentes valores morales, así como mi formación profesional porque sin estimar esfuerzo alguno, han sacrificado gran parte de sus vidas, para formarme y porque nunca podré pagar todos sus desvelos ni aún con la riqueza más grande del mundo. Por lo que soy y por todo el tiempo que les robé pensando en mí. Gracias con todo mi amor y cariño.

A mi adorable madre, le doy gracias por su lucha y esfuerzo que pone cada día, por apoyarme en todo momento, por guiarme y educarme con los mejores principios y valores, por reservarse con tal de darme para mi educación.

A mi padre, por siempre darme ánimos y entender mi cansancio, por aconsejarme cuando más lo necesitaba y su paciencia.

A mis amigos, quienes siempre me levantaron en los peores momentos y me tendieron la mano, quienes me apoyaron en mis tareas confiando y creyendo siempre en mí.

A mis profesores quienes sin dudar confiaron plenamente en mi crecimiento profesional, gracias por compartir sus conocimientos, y brindarme su amistad.

Por ultimo un agradecimiento muy especial al Ing. Miguel Ángel Pérez Solano, a la licenciada Elizabeth y al Ing. Leonardo, quienes siempre me apoyaron en este trayecto, vieron por mí y me apoyaron sin dudarlo, aconsejándome y levantándome los ánimos siempre, gracias por su amistad siempre les estaré eternamente agradecida.

Rosario Elizabeth Velasco Cruz

Agradecimientos

Agradezco a Dios por haberme dado salud, por guiarme en mi camino y por permitirme concluir con mi objetivo. Y así poder culminar un ciclo muy importante en mi vida.

Agradezco a mis padres por acompañarme y brindarme su apoyo incondicional todo este tiempo, que pese a los problemas que se presentaron los pudimos superar juntos, no tengo palabras para agradecerles lo mucho que hicieron por mí, por nunca perder la fe, por creer que podría lograr mis metas, por todas esas noches de desvelo que les provoque.

Gracias a mi madre porque siempre vio por mí, porque siempre estuviera bien, aunque ella no lo estuviera.

Gracias a mi padre que siempre ha trabajado para que yo siguiera estudiando, sacrificando varias cosas con tal de poder darme el estudio.

Agradezco a mis hermanos porque en muchas ocasiones me apoyaron cuando más lo necesitaba, no solo económicamente si no sentimentalmente brindándome consejos que los tengo muy presentes y los llevare conmigo hoy y siempre.

Agradezco a mis amigos con los cuales he pasado demasiados momentos buenos. A lo largo de la carrera fui conociendo personas muy agradables y hoy que esta por culminar esta parte de mi vida agradezco a mis compañeros de residencia ya que han pasado muchas cosas que nos han hechos más fuertes, permitiéndonos salir adelante.

Y por último y no por eso menos importante agradezco a mis profesores, que son personas muy inteligentes y dedicadas a su profesión, gracias a ellos aprendimos muchas cosas que nos servirán en la vida, para desarrollarnos profesionalmente.

Zuriel Juan Vásquez Hernández

Resumen

El presente proyecto de residencia consiste en la búsqueda de servicios turísticos que brinda sugerencias como resultado de los lugares que ya han sido visitados por diferentes turistas anteriormente, punteando en un mapa el lugar, así de este modo se ayuda con la toma de decisiones.

Para la elaboración de las sugerencias de los servicios turísticos se tomaron como referencias encuestas realizadas por prestadores del área de la división de posgrado e investigación del Instituto Tecnológico de Oaxaca.

En el Primer capítulo de dicho documento se describe la parte contextual del proyecto, como es la descripción de la Institución a quien se elabora el sistema, la problemática, los objetivos, justificación, así como los alcances y limitaciones.

En el Segundo capítulo se plantean conceptos que son relacionados con los temas empleados a lo largo del desarrollo del proyecto, además de una comprensión más clara de los temas de las diferentes áreas relacionadas al proyecto.

En el Tercer capítulo se realiza el desarrollo de la metodología que utilizamos para documentar el proyecto realizado, tomando como referencia las herramientas y fases de la metodología XP (Programación Extrema) descritas en el capítulo II de dicho documento, se realizaron tres iteraciones por cada una de ellas se realizó la planeación, el diseño, la codificación y las pruebas.

En la primera iteración se muestran las historias de usuario, el diseño de la base de datos, así como imágenes de las vistas del sistema con su respectiva descripción.

Por último, se encuentra las conclusiones sobre la experiencia que tuvimos con el desarrollo del proyecto y anexos.

ÍNDICE

Resumen 5
Introducción 11
CAPÍTULO I. MARCO CONTEXTUAL 12
1.1 Descripción de la Institución. 13
1.1.1 Organigrama General de la Institución. 14
1.1.2 Organigrama de la División de Estudios de Posgrado e Investigación. 15
1.2 Planteamiento del problema de tesis. 16
1.3 Hipótesis de trabajo 17
1.4 Objetivos 17
1.4.1 Objetivo General 17
1.4.2 Objetivo Específico 17
1.5 Justificación 18
1.6 Alcances y Limitaciones 19
1.6.1 Alcance 19
1.6.2 Limitaciones 19
CAPÍTULO II. MARCO TEÓRICO 20
2.1 Turismo 21
2.1.1 El sistema turístico 21
2.1.2 Demanda turística 22
2.1.3 Turista 22
2.1.4 Oferta Turística 23
2.2 Tipos de turismo 24
2.2.1 Turismo cultural 25
2.2.2 Turismo religioso 25
2.2.3 Turismo gastronómico 25
2.2.4 Turismo idiomático 26
2.2.5 Turismo de salud 26
2.2.6 Turismo de activo 26
2.2.7 Turismo de parques temáticos 26

2.2.8 Turista de negocios 27
2.3 Servicio Turístico 27
2.3.1 Servicios turísticos generales 27
2.3.2 Servicios turísticos privados 28
2.4 Tecnologías de desarrollo web 28
2.4.1 HTML 28
2.4.2 CSS 29
2.4.3 JavaScript 29
2.5 Lenguaje de programación PHP 30
2.5.1 PHP 30
2.6 Paquete de herramientas de desarrollo 31
2.6.1 XAMPP 31
2.7 Base de datos 31
2.7.1 MySQL 31
2.8 Framework 32
2.8.1 CodeIgniter 32
2.8.1.1 Librería tank auth para CodeIgniter 33
2.9 Patrones 34
2.9.1 MVC 35
2.10 Machine Learning 35
2.11 Inteligencia Artificial 36
2.12 Sistemas de recomendación 36
2.12.1 Sistema de recomendación basado en filtrado colaborativo 37
2.12.1.1 Sistema basado en usuarios 38
2.12.1.2 Sistema basado en ítems 40
2.12.1.3 Correlacion de pearson 41
2.13 Programación R 43
2.14 API 43
2.14.1 Api de Google Maps 43
2.15 Metodologías 44
2.15.1 Comparativo entre algunas metodologías 45

2.16 Hipótesis de trabajo 47
2.16.1 Asociativa 47
2.16.2 Atributiva 47
2.16.3 Causal 47
CAPÍTULO III. METODOLOGÍA 48
3.1 XP 49
3.1.1 Características 49
3.2 Herramientas De La Metodología XP 50
3.2.1 Historias de Usuario 50
3.3 Fases de la programación extrema 51
3.3.1 Planeación 51
3.3.2 Diseño 51
3.3.3 Codificación 52
3.3.4 Pruebas 52
CAPÍTULO IV. DESARROLLO DE LA TESIS 54
ITERACIÓN 0 55
4.1 Lista Maestra 55
ITERACIÓN 1 56
4.2 Fase 1: Planeación 56
4.2.2 Historias de Usuario 56
4.3 Fase 2: Diseño 58
4.3.1 Modelo E-R 58
4.3.2 Diagrama relacional 59
4.3.3 Desarrollo de Interfaces 60
4.4 Fase 3: Codificación 63
4.4.2 Codificación Modelo Vista Controlador 63
4.5 Fase 4: Pruebas 68
4.5.1 Ingreso al sistema 68
4.5.2 Registro de nuevos productos 70
4.5.3 Visualización de los productos registrados 70
4.5.3 Actualización de productos turísticos 71

4.5.4 Eliminación de productos turísticos 72
ITERACIÓN 2 74
4.6 Fase 1: Planeación 74
4.6.1 Historias de Usuario 74
4.7 Fase 2: Diseño 76
4.7.1 Diagrama E-R 76
4.7.2 Diagrama relacional 77
4.7.3 Interfaz de Búsqueda de Recomendación de Productos Turísticos 77
4.7.4 Interfaz de Resultados de Recomendación de Productos Turísticos (Hoteles y Restaurantes) 79
4.7.5 Interfaz de Evaluación de Recomendación de Productos Turísticos 79
4.8 Fase 3: Codificación 80
4.8.1 Módulo de Búsqueda de Recomendación de Productos Turísticos 80
3.8.2 Código En R que ejecuta el Algoritmo de Recomendación 89
3.8.3 Módulo de Resultados de Recomendación de Productos Turísticos 92
3.8.4 Módulo de Evaluación de Productos Turísticos 102
4.9 Fase 4 Pruebas 107
ITERACIÓN 3 109
4.9 Fase 1: Planeación 109
3.9.1Historias de usuario 109
4.10 Fase 2: Diseño 110
4.11 Codificación 110
4.11 Pruebas 112
4.12 Pruebas de hipótesis 114
CAPÍTULO V. RESULTADOS Y CONCLUSIONES 116
Resultados de la investigación 117
Conclusión 117
Referencias 118
ANEXOS 121
MANUAL DE USUARIO 121
MANUAL TÉCNICO 141

ÍNDICE DE TABLAS

Tabla 1 Comparación de metodologías 45
Tabla 2 Lista Maestra de Historias de Usuarios. 55
Tabla 3 Historia de Usuario 1. Acceso y Registro a la plataforma 56
Tabla 4 Historia de Usuario 2. Registro de productos turísticos 56
Tabla 5 Historia de Usuario 3. Visualización de productos turísticos 57
Tabla 6 Historia de Usuario 4. Actualización de productos turísticos 57
Tabla 7 Historia de Usuario 8. Eliminación de productos turísticos 57
Tabla 8 Historia de Usuario 6. Visualización de productos turísticos 74
Tabla 9. Historia de Usuario 7. Búsqueda de recomendación de productos turísticos 74
Tabla 10 Historia de Usuario 8. Resultados de recomendación de productos turísticos 75
Tabla 11 Historia de Usuario 9. Evaluación de productos turísticos recomendados 75

Índice de ilustraciones

Ilustración 1. Organigrama General de la Institución 14
Ilustración 2. Organigrama de la División de Estudios de Posgrado e Investigación. 15
Ilustración 3 Amat,J.(2018). Ilustracion Matriz de valoración[Figura]. Recuperado de https://rpubs.com/Joaquin_AR/370301 39
Ilustración 4 Amat,J.(2018). Ilustracion Matriz de valoración[Figura]. Recuperado de https://rpubs.com/Joaquin_AR/370301 41
Ilustración 5. Plantilla Historia de Usuario 50
Ilustración 6 interfaz de sesión 60
Ilustración 7 interfaz registro 60
Ilustración 8 interfaz registro producto 61
Ilustración 9 interfaz visualización de productos turísticos 61
Ilustración 10 interface actualización 62
Ilustración 11 interface eliminación 62
Ilustración 12 prueba inicio sesión 69
Ilustración 13 pantalla administrador 69
Ilustración 14 prueba registro producto 70
Ilustración 15 prueba visualización producto 70
Ilustración 16 prueba listado actualización 71
Ilustración 17 prueba actualización 71
Ilustración 18 prueba listado eliminar 72
Ilustración 19 prueba información a eliminar 72
Ilustración 20 listado eliminación actualizado 73
Ilustración 21 Pantalla de inicio búsquedas 77
Ilustración 22 Formulario de búsqueda 78
Ilustración 23 interfaz de resultados de búsqueda 79
Ilustración 24Interfaz de evaluación 79

Introducción

El sector turístico es uno de los más importantes detonantes de la economía del Estado de Oaxaca, se busca posicionar a Oaxaca como un destino turístico de preferencia en los mercados nacionales e internacionales para incrementar el número de visitantes, su estadía y derrama económica.

Esto puede lograrse mediante el uso de la tecnología, en la actualidad Internet pone ante el usuario tantas opciones que ya no es suficiente con un buscador, ahora se hace uso de tecnología como Inteligencia Artificial o Machine Learning que permite crear algoritmos que replican la lógica que se sigue en la vida antes de tomar una decisión y que lleva a preguntar a un amigo, con el que se comparten gustos, o a un experto de confianza. La gran diferencia es que en lugar de preguntar a dos o tres personas, la inteligencia artificial nos permite hacer consultas que equivaldrían a hablar con una ciudad completa.

Los sistemas de recomendación estudian las preferencias y gustos de los usuarios con el objetivo de recomendarles ítems a comprar o examinar que factiblemente sean de su interés.

Los sistemas de recomendación, como todos los sistemas de inteligencia artificial, se alimentan de datos. El hecho de que hoy la IA sea más tangible, cada interacción en las redes sociales, en la web o en los sistemas de mensajería aportan datos que permiten conocer gustos, tendencias, preferencias, etc.

Un sistema de recomendación tiene dos objetivos principales:

- Sugerir ítems en los que el usuario se espera que esté interesado, y
- Predecir la utilidad de un ítem específico para un usuario en particular.

CAPÍTULO I. MARCO CONTEXTUAL

1.1 Descripción de la Institución.

El Instituto Tecnológico de Oaxaca (ITO) es una institución pública de educación superior localizada en la Ciudad de Oaxaca de Juárez, Oaxaca, México. Forma parte de la Dirección General de Educación Superior Tecnológica (DGEST), de la Secretaría de Educación Pública de México. Imparte 9 carreras a nivel licenciatura y 2 a nivel posgrado en las áreas de ciencias sociales y administrativas, e ingeniería.

Los programas de Doctorado y Maestría en Ciencias en Desarrollo Regional y Tecnológico, son programas orientados a la investigación. Estos dos programas cuentan con tres líneas de investigación: Desarrollo Regional Sustentable, Sistemas Mecánicos y Bioquímica.

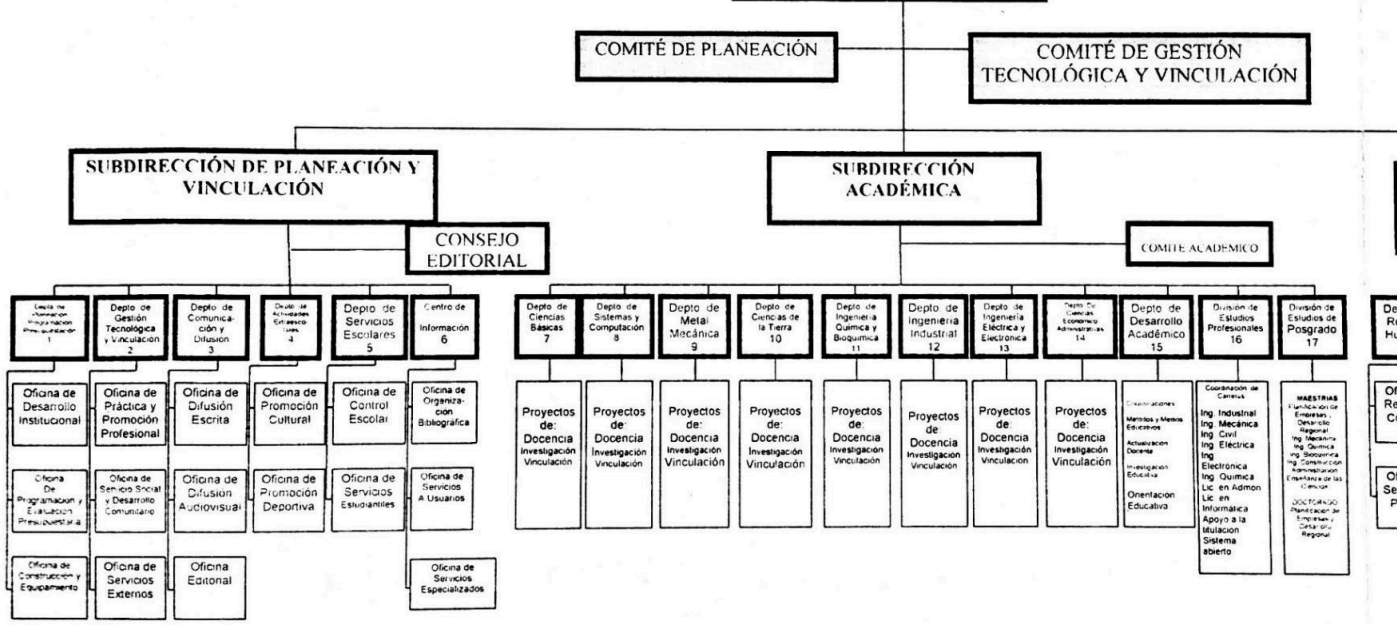
Los programas habilitados de Maestría en Construcción y Maestría en Administración, son programas con orientación profesional que buscan la vinculación con las empresas para desarrollar proyectos de impacto en los sectores de la sociedad.

El programa de Maestría en Docencia es un programa en modalidad no presencial, cuya finalidad es fortalecer el quehacer académico de profesores con actividad docente de los niveles medio y superior.

El equipo de trabajo de profesores y administrativos de la División de Estudios de Posgrado e Investigación, han aceptado el reto de integrar a profesionistas entusiastas a nuestros programas, por lo cual estamos comprometidos con su formación académica y personal.

Los programas de maestría y doctorado se evalúan mediante la realización y entrega de una tesis, además de la publicación de productos de investigación como artículos arbitrados e indexados, la participación a congresos y estancias de investigación en los que los estudiantes se involucran durante sus estudios de posgrado.

1.1.1 Organigrama General de la Institución.



1.1.2 Organigrama de la División de Estudios de Posgrado e Investigación.

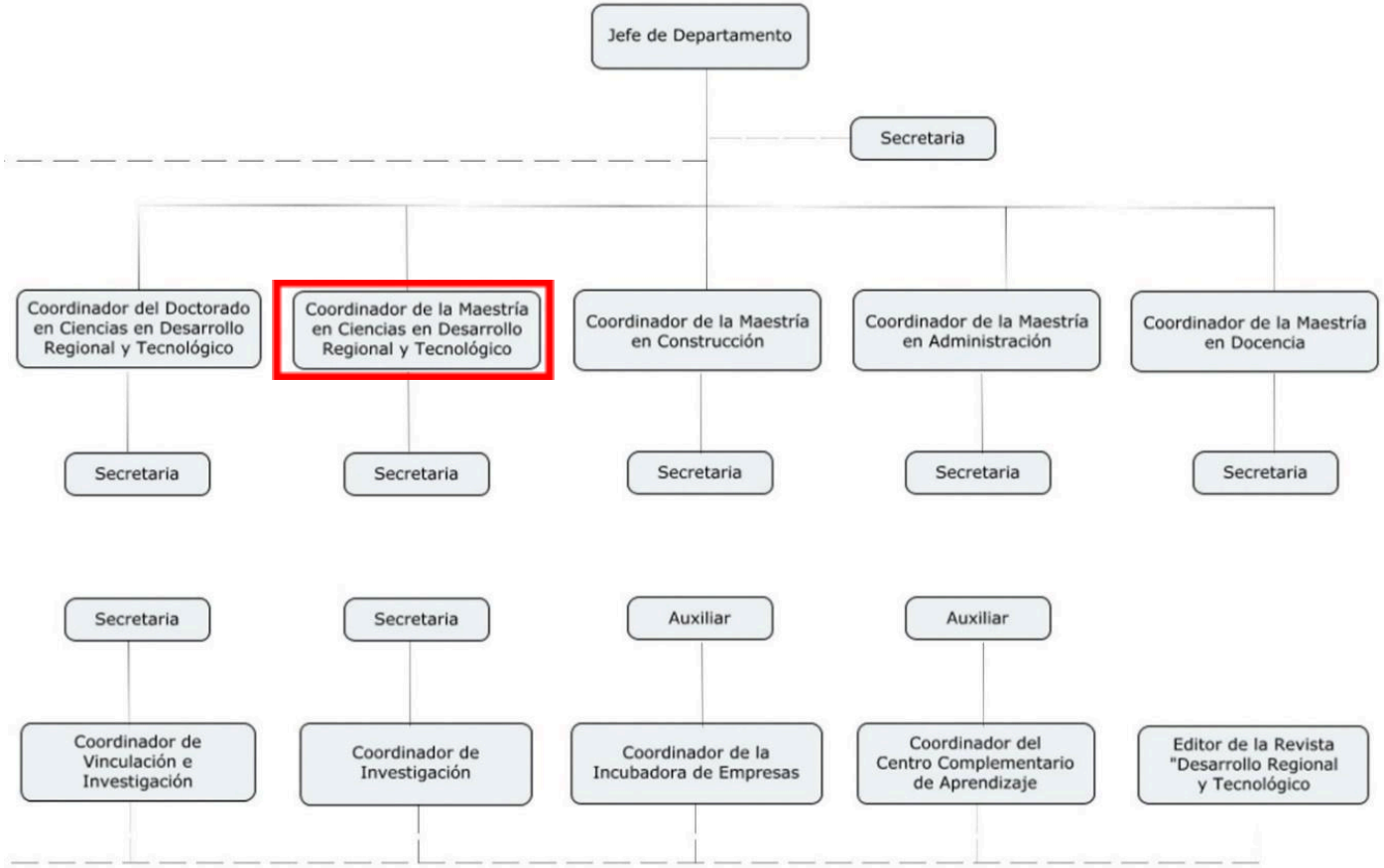


Ilustración 3EQ Ilustración * ARABIC 2. Organigrama de la División de Estudios de Posgrado e Investigación.

Ilustración 3EQ Ilustración * ARABIC 2 Amaz_J.(2018). Ilustración Matriz de valoración[Figura]. Recuperado de https://issuu.com/Joaquin_ARG/docs/Ilustración 3EQ Ilustración * ARABIC 2. Organigrama de la División de Estudios de Posgrado e Investigación.

1.2 Planteamiento del problema de tesis.

El sector turismo en Oaxaca, aún en estos días, presenta un cúmulo de problemas que urge corregir. El desarrollo humano, la gestión, la oferta de servicios turísticos, aspectos culturales y naturales, son sólo algunos de los frentes donde es posible encontrar grandes deficiencias.

En cuanto a cifras turísticas, en Oaxaca se han mantenido entre las más bajas del país durante la última década. Pese a que México recibió en 2017, 35 millones de turistas extranjeros, Oaxaca solo conoció el 0.76% de estos, alrededor de 249 mil Según la Secretaria de Turismo del Estado.

Esta situación se ha generado por diversos motivos, desde estrategias mal planeadas hasta la falta de definición del tipo de lugar que Oaxaca aspira a ser.

Uno de los errores que se ha detectado, es la falta de apoyo a visitantes nacionales y extranjeros para la toma de decisiones que les permita definir destinos turísticos capaces de cumplir con las expectativas deseadas.

En el proceso de planeación, cada turista valora muchos factores antes de elegir un destino. Entre ellos se encuentran los costos de viaje y estadía, los atractivos naturales, la calidad de atención, la oferta recreativa, la calidad de hotelería, gastronomía y seguridad.

Sin embargo, no existe un sistema que permita apoyar una decisión de este tipo ya que a pesar de que se cuenta con información relevante como; calificaciones sobre la calidad de los servicios turísticos que ofrecen los prestadores, comentarios de las experiencias y reseñas, no se cuenta aún con el trabajo en conjunto de toda esta información que permita generar a los turistas varias opciones de destinos.

1.3 Hipótesis de trabajo

Es necesario el desarrollo de una plataforma orientada a fomentar el turismo en Oaxaca que permita el crecimiento y desarrollo del sector turístico apoyando a la toma de decisiones de un turista para elegir las mejores opciones de productos turísticos.

“Sugerir y recomendar productos turísticos a visitantes de acuerdo a su perfil implementando un algoritmo basado en filtrado colaborativo”.

1.4 Objetivos

1.4.1 Objetivo General

Desarrollar una plataforma informática que permita la captación, procesamiento y presentación de información del sector turismo en el estado de Oaxaca, mediante procesos de captura manual y adquisición automática como lo pueden ser aplicación de encuestas, robots de búsqueda y estudios del área de Posgrado.

1.4.2 Objetivo Específico

1. Crear una plataforma íntegra que incluya:
    a. Sistema de captura y búsqueda de productos turísticos mediante parámetros de búsqueda mejorados mediante minería de datos basada en información obtenida de crawlers. (Desktop).
    b. Creación de una Aplicación Desktop para la presentación de los datos de manera ejecutiva, usando Dashboards.
2. Usar la metodología XP
3. Planificar el desarrollo mediante historias de usuario.
4. Diseñar el sistema.
5. Realizar cada una de las iteraciones.
6. Implementar pruebas.
7. Generar documentación.

1.5 Justificación

El desarrollo del turismo tiene múltiples impactos dentro de la sociedad, que van desde lo económico, social y ambiental, por eso desarrollar una plataforma que permita el crecimiento de turismo en el estado es importante. La plataforma tiene la función de ayudar a los turistas a tener opciones que faciliten la toma de decisiones sobre los productos turísticos que pueden elegir dentro del estado de Oaxaca.

Las expectativas de una persona que realiza un viaje a algún destino turístico por lo general son altas, después de haberlo planeado. Por lo cual, con la plataforma de búsqueda el turista podrá preocuparse menos por planear su viaje y enfocarse en descubrir las maravillas turísticas que el estado de Oaxaca tiene que ofrecer.

La plataforma a desarrollar tendrá la funcionalidad de mostrar imágenes atractivas de los lugares, así mismo como sus ubicaciones, para que de este modo, el turista pueda decidir de la mejor manera los lugares que debe y puede visitar.

Considerar que la plataforma trabaja bajo algoritmos de filtrado colaborativo, es una de las ventajas que permite no solo decidir a partir de los datos del turista, sino que toma en cuenta las búsquedas realizada por otros visitantes con el mismo perfil del turista para hacer la sugerencia y recomendaciones de los productos turísticos.

La retroalimentación de la plataforma permite calificar las búsquedas realizadas del turista y generar así nueva información que sirve como referencia para seguir realizando recomendaciones a nuevos turistas que en algún momento decidan utilizar la aplicación

El tiempo que las personas se toman para poder elegir un hotel o un restaurante entre una enorme variedad que existe en la Ciudad de Oaxaca puede llegar a ser muy agotador, es por eso el motivo de la plataforma. Su objetivo más importante es promover los productos

turísticos mejores evaluados, con el mejor servicio, y la mejor calidad que solo Oaxaca puede ofrecer.

1.6 Alcances y Limitaciones

1.6.1 Alcance

- El proyecto se realizará en el área de Posgrado del Instituto Tecnológico de Oaxaca
- Se revisará información en la región de valles centrales de Oaxaca mediante medios digitales.
- Se incluirán los productos turísticos que sean medibles.

1.6.2 Limitaciones

- La plataforma será exclusivamente con acceso vía conexión de internet.
- Se obtendrá información de servicios turísticos de la región de valles centrales de Oaxaca.

CAPÍTULO II. MARCO TEÓRICO

A lo largo de este capítulo se plantean algunos parámetros que sirvan de ejes conceptuales para proporcionar una idea más clara sobre el diseño y desarrollo de dicho proyecto. Por empezar, diseñar las interfaces para ingresar e interactuar los usuarios en la plataforma fue requerido utilizar tecnologías de desarrollo web como son HTML, CSS y JavaScript, los cuales se explica en el punto 2.1.

2.1 Turismo

El turismo es un fenómeno social, cultural y económico relacionado con el movimiento de las personas a lugares que se encuentran fuera de su lugar de residencia habitual por motivos personales o de negocios/profesionales. Estas personas se denominan visitantes y el turismo tiene que ver con sus actividades, de las cuales algunas implican un gasto turístico. (Morillo Moreno, 2011)

El turismo comprende las actividades que realizan las personas durante sus viajes y estancias en lugares distintos al de su entorno habitual, por un período de tiempo consecutivo inferior a un año con fines de ocio, por negocios y otros. (Organizacion Mundial del Turismo, 2011).

2.1.1 El sistema turístico

La naturaleza de la actividad turística es un resultado complejo de interrelaciones entre diferentes factores que hay que considerar conjuntamente desde una óptica sistemática, es decir, un conjunto de elementos interrelacionados entre sí que evolucionan dinámicamente. Concretamente, se distinguen cuatro elementos básicos en el concepto de actividad turística:

- La demanda: formada por el conjunto de consumidores o posibles consumidores de bienes y servicios turísticos. (Organizacion Mundial del Turismo, 2011)

- La oferta: compuesta por el conjunto de productos, servicios y organizaciones involucrados activamente en la experiencia turística. (Organizacion Mundial del Turismo, 2011)

- El espacio geográfico: base física donde tiene lugar la conjunción o encuentro entre la oferta y la demanda y en donde se sitúa la población residente, que, si bien no es en sí misma un elemento turístico, se considera un importante factor de cohesión o disgregación, según se la haya tenido en cuenta o no a la hora de planificar la actividad turística. (Organizacion Mundial del Turismo, 2011)

- Los operadores del mercado: son aquellas empresas y organismos cuya función principal es facilitar la interrelación entre la oferta y la demanda. Entran en esta consideración las agencias de viajes, las compañías de transporte regular y aquellos organismos públicos y privados que, mediante su labor profesional, son artífices de la ordenación y/o promoción del turismo. (Organizacion Mundial del Turismo, 2011)

2.1.2 Demanda turística

Un concepto más estrictamente económico de estas magnitudes haría referencia a la capacidad de gasto –o al gasto real de esas personas–, ya que en términos económicos la demanda es siempre una magnitud monetaria y no el número de consumidores. En este sentido, la OMT (OMT, 1979) en sus definiciones enmarca el concepto de gasto por turismo, el cual define como “el gasto realizado por un visitante o de parte de un visitante por causa de un viaje y durante su estancia en el destino”. (Gurría Di-Bella, 1991)

Turistas, viajeros y visitantes forman la demanda turística y, tras este término, se encuentra un grupo heterogéneo de personas un agregado de personalidades e intereses con diferentes características sociodemográficas, motivaciones y experiencias. Por ello, son varias las clasificaciones que existen dentro de la demanda turística. (Sancho, 2014)

2.1.3 Turista

Una definición muy original de turista es: «La diferencia entre un turista y un viajero es que mientras el primero piensa en regresar desde el mismo momento de su llegada, el viajero puede no regresar nunca», dicho en su novela El Cielo Protector, Paul Bowles (1949). (Sancho, 2014)

Las formas y estilos de vida y los atributos individuales de las personas –actitudes, percepciones, personalidad, experiencias y motivaciones–, dan lugar a diferentes tipos de turistas que Cohen (1974) clasifica en cuatro grandes grupos, describiendo los dos primeros como turismo institucionalizado y los dos últimos como turismo no institucionalizado:
(Sancho, 2014)

1. **El turista de masas organizado:** tiene poco contacto con la cultura local y busca mantener su entorno habitual en el destino de vacaciones.

2. **El turista de masas individual:** similar al anterior, aunque con un poco más de flexibilidad en sus decisiones.

3. **El explorador:** organiza su viaje independientemente, aunque buscando transporte y alojamiento confortables. Abandona el entorno habitual, pero de vez en cuando vuelve a él.

4. **El impulsivo:** no tiene ningún contacto con la industria turística, intenta abandonar su entorno de siempre, no tiene itinerario fijo y tiene contacto con las culturas que visita.

2.1.4 Oferta Turística

El análisis económico entiende por oferta la cantidad de mercancía o servicio que entra en el mercado consumidor a un precio dado y por un periodo dado. Cuando se trata de bienes lo que se produce y no se vende se puede acumular. Esta posibilidad no tiene vigencia en el caso de los servicios, porque cada vez que expira el tiempo de duración de un servicio, su oferta se renueva, y si no se vende, se pierde. Para que un servicio turístico se convierta en oferta turística, es imprescindible que el consumidor potencial conozca su existencia. El turista real consumidor potencial es quien, localmente, establece el tiempo que cada producto puede entrar en el mercado. Asimismo, la oferta turística se agrupa en dos grandes componentes tal como se explica a continuación:
(Reyna M. Ibánez Pérez, 2011)

Infraestructura: Es la dotación de bienes y servicios con que cuenta un país para sostener sus estructuras sociales y productivas. Forman parte de la misma, la educación, los servicios sanitarios, la vivienda, los transportes, las comunicaciones y la energía. Una de las funciones primordiales de la infraestructura es vincular entre sí los asentamientos humanos y resolver las necesidades internas de los mismos, a fin de permitir en esas dos escalas. La circulación de personas, mercaderías, energía etc. El otro tipo de infraestructura es la que corresponde a las dotaciones de la salud, la educación, la vivienda. La otra forma de dividir a la infraestructura en externa o interna, se define por sí misma, es externa porque es general y es general porque sirve a todos los sectores sin pertenecer especialmente a alguno de ellos, aunque a veces pueda beneficiar a unos más que a otros. La infraestructura interna también tiene dos formas: la primera corresponde a la dotación que se da a sí mismo un sector, para operar en una localización precisa. La otra forma es la infraestructura urbana que resume a todas las redes que prestan servicios indistintamente a las diferentes actividades de una ciudad.

Superestructura: La superestructura turística comprende todos los organismos especializados, tanto públicos como privados, encargados de optimizar y cambiar, cuando fuere necesario, el funcionamiento de cada una de las partes que integran el sistema, así como armonizar sus relaciones para facilitar la producción y venta de los múltiples servicios que conforman el sistema turístico. La existencia de una superestructura poderosa y sobre todo

eficiente es mucho más importante de lo que habitualmente se piensa, para que el sector turístico rinda en relación directa la importancia de los atractivos y de la planta que se construyó para explotarlos.

2.2 Tipos de turismo

La actividad turística se clasifica de acuerdo al componente espacial, temporal o incluso de acuerdo al propósito del viaje. Cada modalidad turística engloba un grupo amplio de actividades. (Ibáñez & Rodríguez Villalobos, 2012)

Turismo tradicional. Se le reconoce por su masivo y el desarrollo de grandes instalaciones de alojamiento y espaciamiento. Regionalmente se caracteriza por las visitas a playa o a ciudades coloniales con atractivo culturales en grandes ciudades. (Ibáñez & Rodríguez Villalobos, 2012)

Turismo alternativo. Esta corriente del turismo tiene por objetivo, la realización de viajes, en donde los visitantes participan en actividades recreativas de contacto directo con la naturaleza y las expresiones culturales de comunidades rurales, indígenas y urbanas, siendo respetuosos del patrimonio natural, cultural, e histórico del lugar que vistan. (Ibáñez & Rodríguez Villalobos, 2012)

Esta otra clasificación agrupa actividades de acuerdo a las tendencias de cambio en las necesidades, gustos y preferencias de algunas personas o turistas:

2.2.1 Turismo cultural

Se define como: “Aquel viaje turístico motivado por conocer, comprender y disfrutar el conjunto de rasgos y elementos distintivos, espirituales y materiales, intelectuales y afectivos que caracterizan a una sociedad o grupo social de un destino específico”. En el caso de México el turismo cultural juega un papel muy importante para dar a conocer, preservar y disfrutar el patrimonio cultural y turístico, ya que el tratamiento adecuado del turismo cultural ha generado, desde una perspectiva de mercados, la satisfacción del cliente, la conservación del patrimonio de uso turístico y el desarrollo económico y social de las comunidades a partir de la generación de nuevos empleos. (Reyna M. Ibánez Pérez, 2011)

2.2.2 Turismo religioso

Son desplazamientos de personas por motivos de fervor y devoción religiosa, no importando el sexo, credo o estatus social, ya sea con la finalidad de cumplir un voto o una promesa, pedir algún tipo de beneficio o dar gracias por algún beneficio ya recibido. A diferencia del turismo tradicional, este tipo de turismo hace visitas por ocasiones repetidas al mismo sitio en un periodo menor y, por tanto, resulta más fiel al sitio que visita. (Reyna M. Ibánez Pérez, 2011)

2.2.3 Turismo gastronómico

En la actualidad existen muchos tipos y modalidades de turismo de acuerdo con el interés particular del viajero, es decir, del propósito u objetivo que motiva el viaje; en este sentido, encontramos que hay quienes visitan un destino en particular sólo para conocer y experimentar determinados tipos de comidas y bebidas particulares de la región, para conocer las especialidades gastronómicas que se preparan con motivos o en fechas específicas que coinciden con fiestas cívicas o culturales de importancia local y/o nacional, o porque son de temporada, para conocer sus ingredientes e incluso para aprender a prepararlos, es decir, para descubrir nuevas sensaciones y experiencias culinarias, a eso es a lo que llamamos turismo gastronómico, que está creciendo en el ámbito mundial, paulatinamente. (Reyna M. Ibánez Pérez, 2011)

2.2.4 Turismo idiomático

Se entiende por turismo idiomático al desplazamiento cuya motivación principal es aprender un idioma; en este sentido, el español, inglés, francés y japonés son recursos turísticos con un gran potencial, ya que son idiomas que cada vez adquieren mayor relevancia socioeconómica y se están convirtiendo en una herramienta indispensable para maestros, empresarios, diplomáticos, universitarios, científicos y políticos. (Reyna M. Ibánez Pérez, 2011)

2.2.5 Turismo de salud

Se refiere a las técnicas existentes para el tratamiento de enfermedades o simple relajación en spas, saunas u otros centros de medicina tradicional o alternativa. En países como el nuestro esta actividad genera una derrama importante de divisas, e incluso la permanencia de este tipo de turistas puede ser más prolongada y generar mayor derrama que turistas que vienen de negocios. (Reyna M. Ibánez Pérez, 2011)

2.2.6 Turismo de activo

Está representado por el conjunto de turistas que realizan viajes relacionados con la práctica de algún deporte, desde el más barato y común, hasta el más costoso y sofisticado. Buscan, además, realizar una actividad física que suponga un cambio en la rutina diaria. Por tanto, una característica más de este tipo de turismo es la demanda de actividades que les permitan romper con la rutina diaria, con la monotonía y el estrés. Pero, que, al mismo tiempo, les permitan realizar alguna actividad física.

2.2.7 Turismo de parques temáticos

Se define como el conjunto de turistas que viajan con la finalidad de disfrutar de una atracción o elemento destinado o encaminado al disfrute de las diversiones, entretenimientos y otros recursos de ocio, ya sean naturales o creadas por el hombre, especialmente para atraer visitantes. (Reyna M. Ibánez Pérez, 2011)

2.2.8 Turista de negocios

Se define como el conjunto de corrientes turísticas cuyo motivo de viaje está vinculado con la realización de actividades laborales y profesionales. Puede ser realizado de manera grupal o individual e incluye numerosos segmentos y productos relacionados con la organización de reuniones de negocios, con diferentes propósitos y magnitudes; además, comprende a los viajeros de negocios, que por cuestiones inherentes a sus actividades se desplazan de una ciudad a otra o de un país a otro para desarrollar actividades relacionadas con su profesión, por ejemplo: congresos, convenciones, incentivos y exposiciones. (Reyna M. Ibánez Pérez, 2011)

2.3 Servicio Turístico

Conjunto de actividades perfectamente diferenciadas entre sí, pero íntimamente relacionadas, que funcionan en forma armónica y coordinada con el objeto de responder a las exigencias de servicio planteadas por la composición socioeconómica de una determinada corriente turística. (Alejandra, 2012).

Toda actividad realizada por una persona física o moral, pública o privada, tendiente a satisfacer necesidades específicas directamente planteadas por el desplazamiento turístico. (Alejandra, 2012).

2.3.1 Servicios turísticos generales

Según Miguel Ángel en su obra "Conceptualización, origen y evolución del turismo"(2002) nos comenta que el servicio o los servicios Turísticos:

Son aquellos que se ofrecen en forma gratuita al turista por el simple hecho de serlo, y los proporciona tanto el sector público como el sector privado, y en algunos casos en forma conjunta (Alejandra, 2012).

- Oficinas de información turística.
- Oficinas o centros de convecciones privados, oficiales o mixtos.
- Servicio de tramitación aduanal y migratoria que, en razón de su mayor simplificación, facilitan la internación del turista en general, y de los objetos o enseres necesarios para un mejor disfrute de su estancia.
- Los guías oficiales que se desempañan en recintos culturales y sociales de acceso público, trasmiten valiosa información al turista.

2.3.2 Servicios turísticos privados

Son los que proporcionan generalmente el sector privado o aquellos turistas que reúnen ciertos requisitos, entre los que prevalece, la capacidad de correspondencia a la prestación con una remuneración económica. (Alejandra, 2012)

- **Alojamiento:** Lugar de establecimiento que provee de un espacio para pernoctar, puede ofrecer además servicios complementarios. (Antonietti, 2017)

- **Atracciones:** Conjunto de servicios no incluidos en la actividad principal que brinda el alojamiento turístico. Son empresas de servicios turísticos complementarios los Centros recreativos turísticos. Parques temáticos y aquéllas dedicadas a proporcionar, mediante precio, actividades y servicios para el esparcimiento y recreo de sus clientes, de tipo deportivo, medioambiental, cultural, recreativas o de salud y que reglamentariamente se clasifiquen como tales. (Antonietti, 2017)

- **Restaurantes:** Un restaurante es un comercio en el mayor de los casos, público donde se paga por la comida y bebida, para ser consumidas en el mismo local. Hoy en día existe una gran variedad de modalidades de servicio y tipos de cocina. (Antonietti, 2017)

2.4 Tecnologías de desarrollo web

2.4.1 HTML

Es una tecnología con el que se define el contenido de las páginas web. Mozilla (2005) lo define como el “Lenguaje de Marcado para Hipertextos” (HyperText Markup Language) básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web, como imágenes, listas, vídeos, etc.

(Mozilla.org, 2005)

Determina el contenido de la página web, pero no su funcionalidad. Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript). (Mozilla.org, 2005)

2.4.2 CSS

CSS, por sus siglas en inglés de Hojas de Estilo en Cascada (Cascading Style Sheets), es un lenguaje declarativo que controla el aspecto de las páginas web en el navegador. El navegador aplica declaraciones de estilo CSS a los elementos seleccionados para mostrarlos correctamente. Una declaración de estilo contiene las propiedades y sus respectivos valores, los cuales determinan cómo se verá una página web. (Mozilla.org, 2005)

2.4.3 JavaScript

JavaScript es un lenguaje de scripting multiplataforma y orientado a objetos. Es un lenguaje pequeño y liviano. Dentro de un ambiente de host, JavaScript puede conectarse a los objetos de su ambiente y proporcionar control programático sobre ellos.

Contiene una librería estándar de objetos, tales como Array, Date, y Math, y un conjunto central de elementos del lenguaje, tales como operadores, estructuras de control, y sentencias. (mozilla.org, 2005)

Para codificar las interfaces de usuario se usó el lenguaje de programación web PHP ya que es simple y fácil de usar, para ejecutar y realizar pruebas locales se utilizó un software libre como XAMPP el cual se describe en el punto (2.3), XAMPP fue descargado en la versión 7.2.9 el cual es compatible con la versión 7 de PHP, ya que si alguno de los dos varía con versiones antiguas como la 5 puede causar algún inconveniente por las librerías. En cuanto a los datos obtenidos de las encuestas realizadas por la división de estudios profesionales, se diseñó una base de datos que almacena la información de los servicios turísticos, considerando así MySql como el sistema gestor de base de datos confiable de usar, entre otras características descritas a detalle en el siguiente punto (2.4).

2.5 Lenguaje de programación PHP

2.5.1 PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje de 'scripting' de propósito general y de código abierto que está especialmente pensado para el desarrollo web y que puede ser embebido en páginas HTML. Su sintaxis recurre a C, Java y Perl, siendo así sencillo de aprender. El objetivo principal de este lenguaje es permitir a los desarrolladores web escribir dinámica y rápidamente páginas web generadas; aunque se puede hacer mucho más con PHP. (PHP Group, 2001)

No todos los hostings o alojamientos web soportan PHP ya que requiere que el servidor tenga instalado Internet Information Services (IIS) o Apache con las librerías de PHP. La extensión de los archivos de este lenguaje es ".php". Es uno de los lenguajes de programación web más populares por su rapidez y la facilidad de desarrollo. (PHP Group, 2001)

Ventajas del PHP:

- Una de sus características es la rapidez.
- Lo soportan la mayoría de las plataformas de alojamiento web.
- Puede mezclarse con código HTML, aunque esto dificulta su lectura.
- Puede manejar ficheros y conectarse a distintas bases de datos (MySQL, Oracle, SQL Server, Informix, PostgreSQL, etcétera).
- El software que permite soportarlo en los servidores de hosting es libre y gratuito.
- (PHP Group, 2001)

Desventajas del PHP:

- Para poder ver y testear las páginas que vayamos creando es necesario disponer de un servidor web que soporte PHP.
- Al ser interpretado en el servidor, es más fácil que se colapse cuando el número de peticiones de descarga de páginas aumenta.
- Parte del contenido de las páginas puede no ser accesible a los navegadores, dificultando el posicionamiento de las páginas.

(PHP Group, 2001)

2.6 Paquete de herramientas de desarrollo

2.6.1 XAMPP

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser fácil de instalar y usar. (Apache Friends, 2002)

Cabe aclarar que en muchas páginas web se puede encontrar definido a XAMPP como servidor web, sin embargo, no es así, ya que el servidor web es Apache que es incluido en llámese paquete de herramientas, plataforma o software que atribuye a una licencia libre, sus siglas hacen referencia a la “X” como para el uso en cualquier sistema operativo, “A” al servidor Apache, “M” un sistema gestor de base de datos como actualmente MariaDB o MySQL y las últimas dos “PP” a los lenguajes de programación web PHP y Perl.

2.7 Base de datos

Para Gómez Fuente el término Base de Datos consiste en “una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos, en otras palabras, no es más que un conjunto de información, que se encuentra agrupada o estructurada”. (Gómez Fuentes, 2013)

Según la (Conference of Statisticiens Européens, 1977), es una “Colección de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma particular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones”. (Gómez Fuentes, 2013)

2.7.1 MySQL

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2. MySQL incluye todos 10s elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos. Este

lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos. (Gilfillan, 2003)

Así como las BD tienen una estructura para organizar los datos, también en la parte de codificar se requiere una buena estructura y para ellos utilizamos un Framework llamado CodeIgniter el cual es ligado con el patrón de diseño Modelo Vista Controlador puesto que permite separar el código en tres capas, en el punto 2.4 y 2.5 se presentan algunos conceptos y características de lo antes mencionado.

2.8 Framework

Según Dirk Riehle. (Riehle, 2000) El término Framework es una palabra inglesa, cuya traducción literal es “marco de trabajo”.

Conjunto de clases cooperativas que construyen un diseño reutilizable para un tipo específico de software. Un Framework proporciona la arquitectura partiendo el diseño en clases abstractas y definiendo sus responsabilidades y colaboraciones. Un desarrollador realiza una aplicación haciendo subclases y componiendo instancias a partir de las clases definidas por el Framework. (Riba, 2008)

La palabra framework engloba distintos niveles de abstracción, desde una idea estructural (como lo puede ser por ejemplo el patrón Modelo-Vista-Controlador, que declara separar el código por tres capas bien diferenciadas) hasta un conjunto grande de herramientas y procesos específicos (por ejemplo, Microsoft .NET). La mayoría de los frameworks se basan en los patrones de diseño, los cuales son en gran parte responsables de las mejoras cualitativas que ofrecen los frameworks. (Rodríguez Pérez, 2017)

2.8.1 CodeIgniter

CodeIgniter es un Marco de desarrollo de aplicaciones, un conjunto de herramientas, para las personas que crean sitios web utilizando PHP. Permite desarrollar proyectos mucho más rápido, proporcionando un amplio conjunto de bibliotecas para las tareas más comunes, así como una interfaz simple y una estructura lógica para acceder a estas bibliotecas. (EllisLab, 2002-2018)

Ventajas de utilizar un framework como CodeIgniter

- Las páginas se procesan más rápido, el núcleo de CodeIgniter es bastante ligero.
- Facilidad para crear nuevos módulos, páginas o funcionalidades.
- Acceso a librerías públicas y clases. Entre otras, hay librerías para el login, paginado, calendarios, fechas, etc.
- Estandarización del código. Fundamental cuando hay que tocar código hecho por otra persona o cuando trabaja más de una persona en un mismo proyecto.
- Separación de la lógica y arquitectura de la web, el MVC.
- Cualquier servidor que soporte PHP+MySQL sirve para CodeIgniter.
- CodeIgniter se encuentra bajo una licencia open source, es código libre.

2.8.1.1 Librería tank auth para CodeIgniter

Tank Auth es una biblioteca de autenticación para PHP-framework CodeIgniter . Está basado en DX Auth. (Konyukhov, 2018)

Los puntos clave de la biblioteca son:

Es sencillo

- Opciones básicas de autenticación (inicio de sesión, cierre de sesión, registro, anulación del registro).
- Muy compacto (menos de 20 archivos y 4 tablas de DB).
- El nombre de usuario es opcional, solo el correo electrónico es obligatorio.

(Konyukhov, 2018)

Es seguro

- Usar la biblioteca phpass para el hashing de contraseñas (en lugar de md5 inseguro).
- Contando el intento de inicio de sesión para evitar la fuerza bruta (opcional). Intentos de inicio de sesión fallidos determinados por IP y por nombre de usuario.
- Registro de la última dirección IP y hora de inicio de sesión (opcional).
- CAPTCHA para el registro y el intento de inicio de sesión repetitivo (opcional).

- Las cuentas no activadas y las solicitudes de contraseña olvidadas caducan automáticamente.

(Konyukhov, 2018)

Es fácil de administrar

- Modelo estricto MVC: controlador para controlar, vistas para representación y biblioteca como interfaz de modelo.
- Soporte de archivos de idioma.
- Los archivos de vista contienen solo el código HTML necesario sin decoración redundante.
- La mayoría de las características son opcionales y pueden ajustarse o desconectarse en un archivo de configuración bien documentado.

(Konyukhov, 2018)

Tiene todas las funciones

- Inicie sesión usando el nombre de usuario, la dirección de correo electrónico o ambos (dependiendo de la configuración).
- El registro es instantáneo o después de la activación por correo electrónico (opcional).
- La opción "Recordarme".
- Olvidé mi contraseña (permitiendo que los usuarios elijan una nueva contraseña al reactivarla).
- Cambiar contraseña o correo electrónico para usuarios registrados.
- El correo electrónico o la contraseña se pueden cambiar incluso antes de que se active la cuenta.
- Prohibir usuario (opcional).
- Perfil de usuario (opcional).
- Soporte de CAPTCHA (CI-native y reCAPTCHA están disponibles).
- HTML o correos electrónicos de texto sin formato.

(Konyukhov, 2018)

2.9 Patrones

Los patrones de diseño son técnicas que se utiliza para resolver problemas, es decir, es una propuesta o solución que alguien ya planteo para determinado tipo de problema. (Luján, s.f.)

2.9.1 MVC

El patrón Model - View - Controller (Modelo - Vista - Controlador). Este concepto fue acuñado por primera vez por Trygve Reenskaug, un lenguaje antiguo orientado a objetos. Es un patrón de diseño basado en la estructuración y en la reutilización de código, donde los componentes de un sistema son separados por la función que realizan. (Para programadores, 2018)

Esta clasificación se compone, como su nombre indica, de tres partes bien diferenciados:

- Modelo. Representa y especifica la lógica de negocio y el tratamiento de datos, ya sea inserción, extracción o modificación, entre otros. Este se encarga de suministrar, ya sea al controlador o a la vista, los datos reclamados por el controlador.
- Vista: Describe el diseño y la apariencia de la interfaz que va a mostrar el sistema. Muestra y formatea los datos procedentes del modelo y retroalimenta al usuario cualquier tipo de información.
- Controlador: se encarga de recibir las órdenes del usuario y de ejecutarlas. Organiza el flujo de trabajo del sistema.

Los patrones no solo son útiles en estructurar código como el caso de MVC, los patrones los podemos encontrar en diversos ámbitos, como el Machine Learning o bien el aprendizaje automático, es un término asociado a la Inteligencia Artificial que trata sobre el estudio de sistemas que pueden aprender a partir del análisis de los datos que reciben, en relación con lo anterior se planeó el desarrollo de la búsqueda de servicios turísticos para que el sistema de sugerencias mediante algoritmos de sistemas de recomendación los cuales se explican en los siguientes puntos (2.6 y 2.7). (Para programadores, 2018)

2.10 Machine Learning

El aprendizaje automático o Machine Learning utiliza una variedad de algoritmos que aprenden iterativamente de datos para mejorar, describir datos y predecir resultados.

A medida que los algoritmos ingieren datos de entrenamiento, es posible producir modelos más precisos basados en esos datos. Un modelo de aprendizaje automático es la salida generada cuando entrena su algoritmo de aprendizaje automático con datos. Después del entrenamiento, cuando se proporciona un modelo con una entrada, se le dará una salida. Por ejemplo, un algoritmo predictivo creará un modelo predictivo. Luego, cuando proporciona datos del modelo predictivo, recibirá una predicción basada en los datos que entrenaron al modelo. (Hurwitz & Kirsch, 2018)

El aprendizaje automático generalmente se refiere a los cambios en los sistemas que realizan tareas. Asociado a la inteligencia artificial (IA). Dichas tareas implican reconocimiento, diagnóstico, planificación, control de robots, predicción, etc. Los "cambios" pueden ser mejoras a los sistemas que ya están funcionando o la síntesis de nuevos sistemas. Para ser un poco más específico, mostramos la arquitectura de una IA típica. (Stanford University, 1998)

2.11 Inteligencia Artificial

La Inteligencia Artificial (IA) es la rama de la ciencia que se encarga del estudio de la inteligencia en elementos artificiales y, desde el punto de vista de la ingeniería, propone la creación de elementos que posean un comportamiento inteligente. Dicho de otra forma, la IA pretende construir sistemas y máquinas que presenten un comportamiento que, si fuera llevado a cabo por una persona, se diría que es inteligente. El aprendizaje, la capacidad de adaptación a entornos cambiantes, la creatividad, etc., son facetas que usualmente se relacionan con el comportamiento inteligente. (ROMERO, DAFONTE, GÓMEZ, & PENOUSAL, 2007)

2.12 Sistemas de recomendación

Los sistemas de recomendación se encargan de proporcionar a los usuarios consejos e información personalizada sobre productos o servicios que puedan ser de interés a la hora de tomar una decisión. Este proceso, en el que el sistema guía al usuario a la hora de realizar una elección, puede proporcionar resultados que sean de gran utilidad, ya sea ahorrando tiempo, proporcionando datos relevantes de forma cómoda y fácil, e incluso obteniendo información que permite valorar opciones que de otra forma antes no se habrían contemplado, algo muy apreciado por la mayoría de los usuarios.

Un sistema de recomendación reduce su problemática a predecir la puntuación que el usuario daría a una serie ítems que todavía no ha puntuado. Esta predicción, de la forma más intuitiva, podría basarse en las puntuaciones que otros usuarios asignaron en el pasado (sistemas colaborativos o sistemas de filtrado colaborativo), aunque también puede obtenerse usando otra clase de información distinta a las valoraciones mencionadas, como pueden ser las características concretas de los productos, los perfiles de preferencias de los distintos usuarios a la hora de tomar decisiones, información demográfica sobre los usuarios, heurísticas basadas en comportamientos humanos, etc., y por supuesto, usar combinaciones de diversas aproximaciones de entre las existentes. (Castellano Torres, 2007)

2.12.1 Sistema de recomendación basado en filtrado colaborativo

Los sistemas basados en filtrado colaborativo trabajan recogiendo juicios humanos, expresados como votaciones de cada individuo para una serie de ítems en un domino dado, y tratan de emparejar personas que comparten las mismas necesidades o gustos.

Los usuarios de un sistema colaborativo comparten sus valoraciones y opiniones con respecto a los ítems que conocen de forma que otros usuarios del sistema puedan decidir qué elección realizar. A cambio, el sistema proporciona recomendaciones personalizadas para aquellos elementos que resultan interesantes para el usuario.

Es importante tener en cuenta que en el filtrado colaborativo son los usuarios, las personas, quienes determinan la relevancia, cualidad e interés de los ítems, por lo que se puede realizar el filtrado sobre elementos difíciles de analizar mediante computación. Esto se consigue gracias a que el problema no se intenta solucionar analizando los elementos que tratamos de recomendar, sino manejando directamente las valoraciones, objetivas y/o subjetivas, que sobre esos elementos se han realizado.

Además, tiene la capacidad de discernir cómo se adapta un ítem a las necesidades o intereses de los usuarios, basándose en la propia capacidad de los humanos de analizar en términos de calidad o gusto, algo difícil de realizar por procesos computacionales.

Los algoritmos de recomendación colaborativos pueden ser agrupados en dos clases generales:

(Castellano Torres, 2007)

- **Algoritmos basados en memoria** (o en vecindad, o heurísticos)

Los algoritmos basados en memoria son esencialmente heurísticas que realizan predicciones basadas en una colección completa de ítems valorados previamente por el usuario. Es decir, el valor de una puntuación no conocida *v* para un usuario *u* sobre un ítem *i* se calcula como un agregado de las valoraciones de otros usuarios (generalmente, los *K* más parecidos) para el mismo ítem *i*.

- **Algoritmos basados en modelos**

Las aproximaciones basadas en modelos utilizan la colección de valoraciones para aprender un modelo, el cual será utilizado a la hora de realizar las futuras predicciones. Algunos algoritmos colaborativos basados en modelos se fundamentan en:

- Uso de redes bayesianas y derivados.
- Métodos de clustering para filtrado colaborativo.
- Análisis iterativo de la componente principal.
- Otro método interesante es utilizar modelos de aprendizaje de forma que se pueda tratar el filtrado colaborativo como un problema de clasificación.

Para aplicar el filtrado colaborativo y llegar a sugerir los servicios turísticos fue necesario aplicar correctamente el algoritmo, comenzando con la matriz de ítems y perfiles los cuales fueron agregados en R un lenguaje de programación estadístico (apartado 2.9) para realizar los cálculos del valor de correlación Pearson o similitud y posteriormente aplicar la formula parar

generar las predicciones del usuario. Posterior al generar las sugerencias de los servicios turísticos, estos son punteados en un mapa mediante la API de Google Maps(descrita en el apartado 2.10) para orientar a los turistas sobre las rutas y ubicaciones en las cuales puede desplazarse para llegar a su destino.

2.12.1.1 Sistema basado en usuarios

Los sistemas de filtrado colaborativo basado en usuarios predicen la valoración que un determinado usuario hará sobre un producto utilizando las valoraciones que han hecho sobre ese mismo producto los **n** usuarios más parecidos a él. La similitud entre usuarios se mide acorde al patrón de valoraciones que tiene cada uno, en este caso, las filas de la matriz.
(Amat Rodrigo, 2018)

En primer lugar, se calcula la similitud entre el usuario **ux** y el resto de usuarios. En este caso, se emplea como medida de similitud el coeficiente de correlación de Pearson. Como los usuarios están definidos por las filas, hay que transponer la matriz para realizar los cálculos. (Amat Rodrigo, 2018)

```
##   Usuario_A Usuario_B       corr
## 1       u_x       u_1  0.8528029
## 2       u_x       u_2  0.7071068
## 3       u_x       u_3  0.0000000
## 4       u_x       u_4 -0.7921180
```

Una vez ordenados los usuarios de mayor a menor similitud respecto al usuario **ux**, se procede a calcular la predicción de la valoración. Existen varias formas de hacerlo:
(Amat Rodrigo, 2018)

- Promedio las valoraciones (Rjk) de los n usuarios más cercanos. Con esto, se evita tener en cuenta la valoración de usuarios que tienen un perfil muy distinto del usuario de interés. n se debe considerar como un hiperparámetro cuyo valor óptimo se identifica, por ejemplo, mediante validación cruzada. Véase el resultado de este problema si se emplea n = 3. Los 3 usuarios más similares a **ux** son : **u1, u2, u3**. La predicción de la valoración que hace el usuario **ux** sobre el ítem i5 se obtiene como la media de las

valoraciones que cada uno de los usuarios seleccionados tiene sobre el ítem 5 (Rusuario,item5).

$$\textbf{Predicción(ux,i5)=}Ru1,i5+Ru2,i5+Ru3,i53$$

(Amat Rodrigo, 2018)

- El inconveniente de la aproximación anterior es que los **n** usuarios seleccionados tienen el mismo peso en la predicción, sin embargo, no todos se parecen en la misma medida al usuario de interés. Una forma de compensar esta influencia es ponderando la media con los valores de similitud, de esta forma, la valoración de los usuarios pesa más cuanto más se parecen al usuario estudiado. Esta estrategia solo puede aplicarse cuando la similitud toma valores en el rango [0, número positivo], ya que, la media aritmética ponderada, no está definida para pesos negativos y, al menos uno de los pesos, debe ser mayor de cero. Otra opción es considerar las similitudes negativas como 0 de forma que no contribuyen en el cálculo.

$$\textbf{Predicción(ux,i5)=}\frac{corr(ux,u1)\times Ru1,i5+corr(ux,u2)\times Ru2,i5+corr(ux,u3)\times Ru3,i5}{corr(ux,u1)+corr(ux,u2)+corr(ux,u3)}$$

(Amat Rodrigo, 2018)

- Los ítems que tienen siempre valoraciones muy positivas aportan poca información sobre el perfil de usuarios. Por ejemplo, un reloj que esté de oferta al precio de 1 euro cuando realmente vale 100, posiblemente sea bien valorado por todos los usuarios independientemente de su perfil de preferencias. Para evitar este problema se pueden pesar los ítems en función de su varianza en el momento que se calculan las similitudes entre perfiles, o eliminar aquellos ítems con varianza próxima a cero.

(Amat Rodrigo, 2018)

2.12.1.2 Sistema basado en ítems

La idea es muy similar al método basado en usuarios, pero en este caso, se identifican ítems similares (empleando el perfil de valoraciones que han recibido) en lugar de usuarios similares. Además, los ítems que participan en el proceso tienen que haber sido valorados por el usuario de interés **ux**. (Amat Rodrigo, 2018)

En primer lugar, se calcula la similitud entre el ítem i5 y el resto de ítems. En la matriz datos, se corresponde con la similitud entre columnas. En este ejemplo, se emplea como medida de similitud el coeficiente de correlación de Pearson. (Amat Rodrigo, 2018)

```
##   Item_A Item_B       corr
## 1    i_5    i_1  0.9694584
## 2    i_5    i_4  0.5816751
## 3    i_5    i_3 -0.4276180
## 4    i_5    i_2 -0.4780914
```

Una vez calculadas las similitudes entre el ítem i5 y el resto, se seleccionan los n ítems más parecidos y se obtiene la predicción a partir de las valoraciones que el usuario ux ha hecho de esos n ítems. (Amat Rodrigo, 2018)

- Predicción basada en el promedio de los n = 3 ítems más parecidos (i1, i4, i3):

$$\textbf{Predicción(ux,i5)}=Rux,i1+Rux,i4+Rux,i33$$

- Predicción basada en el promedio ponderado por similitud:

$$\textbf{Predicción(ux,i5)}=\frac{corr(i5,i1)\times Rux,i1+corr(i5,i4)\times Rux,i4+corr(i5,i3)\times Rux,i3}{corr(ux,i1)+corr(ux,i4)+corr(ux,i3)}$$

(Amat Rodrigo, 2018)

2.12.1.3 Correlacion de pearson

El coeficiente de correlación de Pearson, pensado para variables cuantitativas (escala mínima de intervalo), es un índice que mide el grado de covariación entre distintas variables relacionadas linealmente. Adviértase que decimos "variables relacionadas linealmente". Esto significa que puede haber variables fuertemente relacionadas, pero no de forma lineal, en cuyo

caso no proceder a aplicarse la correlación de Pearson. Por ejemplo, la relación entre la ansiedad y el rendimiento tiene forma de U invertida; igualmente, si relacionamos población y tiempo la relación será de forma exponencial.

El coeficiente de correlación de Pearson es un índice de fácil ejecución e, igualmente, de fácil interpretación. Digamos, en primera instancia, que sus valores absolutos oscilan entre 0 y 1. Esto es, si tenemos dos variables **X** e **Y**, y definimos el coeficiente de correlación de Pearson entre estas dos variables como r_{xy} entonces:

$$\mathbf{0 \leq} r_{xy} \mathbf{\leq 1}$$

Hemos especificado los términos "valores absolutos" ya que en realidad si se contempla el signo el coeficiente de correlación de Pearson oscila entre –1 y +1. No obstante ha de indicarse que la magnitud de la relación viene especificada por el valor numérico del coeficiente, reflejando el signo la dirección de tal valor. En este sentido, tan fuerte es una relación de **+1** como de **-1**. En el primer caso la relación es perfecta positiva y en el segundo perfecta negativa.

¿Cómo se calcula?

Para contar con una guía que permita:

- Establecer la variación contigua de las dos variables
- Comparar los distintos casos entre sí

Para ello se hace uso del coeficiente de correlación de Pearson, definido como la covarianza que se da entre dos variables tipificadas y se calcula con la siguiente expresión:
(Riquelme, 2014)

$$r_{xy} = \frac{\Sigma Z_X Z_Y}{N}$$

¿Cómo ese interpreta el coeficiente de correlación de Pearson?

Su dimensión indica el nivel de asociación entre las variables.

- Cuando es menor a cero (r < 0) Se dice que hay correlación negativa: Las variables se correlacionan en un sentido inverso.

A valores altos en una de las variables, le suelen corresponder valores bajos en la otra variable y viceversa. Cuánto el valor esté más próximo a -1 dicho coeficiente de correlación más evidente será la covariación extrema. (Riquelme, 2014)

Si r= -1 se habla de correlación negativa perfecta, la cual supone una determinación absoluta entre ambas variables, en sentido directo coexiste una relación lineal perfecta de pendiente negativa.

- Cuando es mayor a cero (r > 0) Se dice que hay correlación positiva: Ambas variables se correlacionan en un sentido directo.

  A valores altos en una de las variables, le corresponden valores altos en la otra variable e igualmente en una situación inversa sucede con los valores bajos. Cuánto más próximo a +1 se encuentre el coeficiente de correlación más evidente será la covariación. Si r = 1 Se habla de correlación positiva perfecta, la cual supone una determinación absoluta entre las variables, en sentido directo coexiste una relación lineal perfecta de pendiente positiva). (Riquelme, 2014)

- Cuando es igual a cero (r = 0) Se dice que las variables están incorrectamente relacionadas, no puede es posible establecer algún sentido de covariación.

  No existe relación lineal, pero esto no implica necesariamente que las variables sean independientes, pudiendo existir relaciones no lineales entre las variables.

  Cuando las dos variables son independientes se dice que están correlacionadas, aunque el resultado de reciprocidad no es necesariamente cierto. (Riquelme, 2014)

2.13 Programación R

El lenguaje de desarrollo R sirve para analizar, procesar y manipular datos estadísticos, así como gráficos. (Contreras García , Molina Portillo, & Arteaga Cezón, s.f.)

Algunas características de R son:

- Almacenar y manipular datos.
- Operadores para calcular sobre variables indexadas (Arrays), en particular matrices.
- Herramientas para analizar datos.
- Posibilidades gráficas para analizar datos.

(Contreras García , Molina Portillo, & Arteaga Cezón, s.f.)

2.14 API

Una API, siglas de Application Programming Interface o Interfaz de Programación de Aplicaciones, es un conjunto de código que se puede emplear para que varias aplicaciones se comuniquen entre ellas. Es algo que realiza una tarea similar a la interfaz de usuario a la hora de fomentar la interacción entre persona y programa, solo que aplicado única y exclusivamente dentro del entorno del software. (NeoAttack, 2018)

2.14.1 Api de Google Maps

El Api de Google Maps permite personalizar mapas con su propio contenido e imágenes para mostrarlos en páginas web y dispositivos móviles. La API de JavaScript de Google Maps presenta cuatro tipos de mapas básicos (hoja de ruta, satélite, híbrido y terreno) que se puede modificar utilizando capas y estilos, controles y eventos, diversos servicios y bibliotecas. (Google Maps Platform, 2018)

En cuanto a estructurar el documento, así desarrollar el proceso de dicho proyecto fue requerida una metodología que cumpliera con los parámetros necesarios, por ello se definen a continuación algunas de las metodologías agiles más conocidas y la selección de la misma.

2.15 Metodologías

Las Metodologías Ágiles resuelven los problemas surgidos, posteriormente, a la masificación del uso del computador personal, dado que las expectativas y necesidades por parte de los usuarios se hicieron más urgentes y frecuentes. (Orjuela Duarte & Rojas C., 2008)

Entre las principales metodologías ágiles se encuentran las siguientes:

- **Programación eXtrema (XP):** La Programación Extrema o Extreme Programing, es un enfoque de la ingeniería de software formulado por Kent Beck, se considera el más destacado de los procesos ágiles de desarrollo de software. Al igual que estos, la

programación extrema se diferencia de los métodos tradicionales principalmente en que presenta más énfasis en la adaptabilidad que en la previsibilidad.

- **Scrum:** Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos: El Desarrollo del software se realiza mediante iteraciones denominadas Sprint, con una duración de 30 días, el resultado de cada Sprint es un incremento ejecutable que se muestra al cliente. La Segunda Característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

- **Metodologías de Cristal:** Se Trata de un conjunto de metodologías para el desarrollo de software caracterizadas por encontrarse centradas en las personas que componen equipo y la reducción al máximo del número de artefactos producidos. El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas. Estas políticas dependerán del tamaño del equipo, estableciéndose una clasificación por colores, por ejemplo: Crystal Clear (3 a 8 integrantes), Crystal Orange (25 a 50 integrantes).

- **Desarrollo de software adaptable (ASD):** Presupone que las necesidades del cliente son cambiantes. La Iniciación de un proyecto involucra definir una misión para él, determinar las características, las fechas y descomponer el proyecto en una serie de pasos individuales, cada uno de los cuales puede abarcar entre cuatro y ocho semanas. Los pasos iniciales deben verificar el alcance del proyecto, los tardíos tienen que ver con el diseño de la arquitectura, la construcción del código, la ejecución de las pruebas finales y el despliegue.

    - **Proceso unificado de desarrollo de software:** El Proceso Unificado, es un “Conjunto de actividades necesarias para transformar los requisitos del usuario en un sistema software”. Este proceso se repite a lo largo de una serie de ciclos

que constituyen la vida de un sistema. Cada ciclo consta de cuatro fases como son: Inicio, Elaboración, Construcción y Transición.

(Castellano Torres, 2007)

2.15.1 Comparativo entre algunas metodologías

Tabla 1
Comparación de metodologías

	XP	SCRUM	RUP	RAD
Tipo de Frame-work	Basado en la adaptabilidad , mayor, flexibilidad, dinámico y funcional	Gestión y desarrollo de software, basado en un proceso iterativo e incremental.	Análisis, diseño, implementación y documentación de sistemas orientados a objetos.	Desarrollo interactivo, construcción de prototipos y el uso de utilizadas CASE.
Tipo de revisión	Se debe integrar como mínimo una vez al día, y realizar las pruebas sobre la totalidad del proceso.	Breve revisión diaria dónde se describen tres cuestiones: -Trabajo realizado el día anterior. -Trabajador previsto a realizar. -Cosas que pueda realizar o impedimentos.	En cada fase se realiza una o más iteraciones, perfeccionando así los objetos. Si no termina una fase no se continua con la siguiente.	Sus pruebas se realizan al finalizar el proceso, enfatizado en la reutilización de los componentes de los programas ya comprobados.
Objetivos	Basadas en dar prioridad a trabajos con resultado directo. -Satisfacción al cliente. -Trabajo en grupo -Actuar sobre variables: Coste, tiempo, calidad y alcance.	Indicado para proyectos en entornos complejos: -Obtener resultados prontos. -Requisitos cambiantes. -Innovación y competitividad fundamentales.	Orientado a objetos que estable las bases, plantillas y ejemplos para todos los aspectos y fases de desarrollo de software.	-Crear o redefinir modelos existentes. -Alto rendimiento, ahorro de tiempo. -Reducción de costos de desarrollo. -Mantener la calidad en todo el desarrollo.

TIPOS DE DESA-RROLLO	Liviana y adaptable. Desarrollo por fases. -Planificación -Diseño -Codificación -Pruebas.	-Desarrollo simple, qué requiere trabajo duro. -Control de forma empírica y adaptable a la evolución del proyecto.	Proceso iterativo incremental por fases: -Inicio -Elaboración -Construcción -Transportación	Desarrollo iterativo por fases: -Modelo de gestión. -Modelo de datos. -Modelo de procesos -Generación de aplicaciones.
FACILIDAD DE USO	-Orientada para pequeños o medianos equipos. -Proyectos de riesgo: Fecha de entrega. -No apto para mucho personal. -Requisitos con probabilidad de cambiar.	-No se basa en el seguimiento de un plan. -Modelo adaptable. -Construcción incremental basadas en iteraciones. -No existe trabajo con diseños o atracciones.	-Dirigido por casos de uso. -Establecimiento temprano de una buena arquitectura. -Iterativo e incremental. -Incremental, el trabajo se divide en mini proyectos.	-Usado para estructurar, Planificar y controlar -Participación activa en talleres con profesionales. -Trabajar cerca con los usuarios.

NOTA: Recuperado de "Estudio comparativo de Metodologías de desarrollo de software", de Arteaga, J. S.f de Enero, 2019. Recuperado de **http://biblioteca.udenar.edu.co:8085/atenea/biblioteca/90255.pdf**

La metodología XP fue posible adaptarla a nuestras necesidades en contraste con otras metodologías que puede observar en la tabla 1, debido a que se puede trabajar en equipo de más de dos personas, brindar facilidad para hacer cambios en el proyecto si es requerido por el cliente, así como desarrollar con facilidad las iteraciones que conllevan sus cuatro fases.

2.16 Hipótesis de trabajo

La hipótesis de trabajo es la que sirve para intentar demostrar una relación concreta entre variables a través de un estudio científico. Estas hipótesis se verifican o se refutan por medio del método científico, por lo que en ocasiones también se conocen como "hipótesis operacionales". Generalmente, las hipótesis de trabajo nacen de la deducción: a partir de ciertos principios generales, el investigador asume ciertas características de un caso particular. Las hipótesis de trabajo tienen varios subtipos: asociativas, atributivas y causales.

2.16.1 Asociativa

La hipótesis asociativa concreta una relación entre dos variables. En este caso, si conocemos el valor de la primera variable, podemos predecir el valor de la segunda.

2.16.2 Atributiva

La hipótesis atributiva es la que se usa para describir los hechos que ocurren entre las variables. Se usa para explicar y describir fenómenos reales y mensurables. Este tipo de hipótesis solo contiene una variable.

2.16.3 Causal

La hipótesis causal establece una relación entre dos variables. Cuando una de las dos variables aumenta o disminuye, la otra sufre un aumento o disminución. Por tanto, la hipótesis causal establece una relación causa-efecto entre las variables estudiadas. Para identificar una hipótesis causal se debe establecer un vínculo de causa efecto, o relación estadística (o probabilística). También es posible verificar esta relación a través de la refutación de explicaciones alternativas. Estas hipótesis siguen la premisa: “Si X, entonces Y”.

CAPÍTULO III. METODOLOGÍA

Para documentar el desarrollo de dicho proyecto fue utilizada la metodología de programación extrema (XP) ya que brinda facilidad para hacer cambios en el proyecto si es requerido por el cliente, también de adaptar el equipo de trabajo de más de 2 personas, dicha metodología cuenta con 4 fases para su desarrollo: Planeación, diseño, codificación y pruebas.

Las 4 fases antes mencionadas se repiten en cada proceso al que se le llama iteración, en cada iteración se muestra las actividades realizadas. Una de las herramientas que fue usada de esta metodología son las historias de usuario, en ella se describe el comportamiento del sistema que el cliente requiere.

3.1 XP

Es una Metodología ligera de desarrollo de aplicaciones que se basa en la simplicidad, la comunicación y la realimentación del código desarrollado. (UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA, 2016)

Objetivos de XP (Programación Extrema):

- La Satisfacción del cliente.
- Potenciar el trabajo en grupo.
- Minimizar el riesgo actuando sobre las variables del proyecto: costo, tiempo, calidad, alcance.

(UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA, 2016)

3.1.1 Características

- Metodología basada en prueba y error para obtener un software que funcione realmente.
- Está orientada hacia quien produce y usa software (el cliente participa muy activamente).
- Reduce el coste del cambio en todas las etapas del ciclo de vida del sistema.
- Cliente bien definido.
- Los requisitos pueden cambiar.

- Grupo pequeño y muy integrado (2-12 personas).
- Equipo con formación elevada y capacidad de aprender.

(UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA, 2016)

3.2 Herramientas De La Metodología XP

3.2.1 Historias de Usuario

Representar las historias de Usuario es realizar una breve descripción del comportamiento del sistema, se realizan por cada característica principal del sistema y son utilizadas para cumplir estimaciones de tiempo y el plan de lanzamientos, así mismo reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación.

Cada historia de usuario debe ser lo suficientemente comprensible y delimitada para que los programadores puedan implementarlas en unas semanas.

La Plantilla a utilizarse para la elaboración de las historias de usuario se muestra en la imagen **3.** (UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA, 2016)

HISTORIA DE USUARIO	
Número: Permite identificar a una historia de usuario.	**Usuario:** Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario.
Nombre Historia: Describe de manera general a una historia de usuario.	
Prioridad en Negocio: Grado de importancia que el cliente asigna a una historia de usuario.	**Riesgo en Desarrollo:** Valor de complejidad que una historia de usuario representa al equipo de desarrollo.
Puntos Estimados: Número de semanas que se necesitará para el desarrollo de una historia de usuario.	**Iteración Asignada:** Número de iteración, en que el cliente desea que se implemente una historia de usuario.
Programador Responsable: Persona encargada de programar cada historia de usuario.	
Descripción: Información detallada de una historia de usuario.	
Observaciones: Campo opcional utilizado para aclarar, si es necesario, el requerimiento descrito de una historia de usuario.	

Ilustración 5. Plantilla Historia de Usuario
Calero,A. (2016). METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION EXTREMA. [Ilustración]. Recuperado de http://repositorio.unan.edu.ni/1365/1/62161.pdf

3.3 Fases de la programación extrema

Acorde con Roger Pressman dice que “La programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas”. (S. Pressman, 2010)

3.3.1 Planeación

La Metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores. El proyecto comienza recopilando las historias de usuarios. Una vez obtenidas estas historias de usuarios, los programadores evalúan rápidamente el tiempo de desarrollo de cada una.
(S. Pressman, 2010)

Los Conceptos básicos de la planificación son:

- Las Historias de Usuarios, las cuales son descritas por el cliente.

- El Plan de Entregas (Release Plan), establece que las historias de usuarios serán agrupadas para conformar una entrega y el orden de las mismas.

- Plan de Iteraciones (Iteration Plan), las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.

- Reuniones Diarias de Seguimiento (Stand – Up Meeting), el objetivo es mantener la comunicación entre el equipo y compartir problemas y soluciones.

3.3.2 Diseño

El diseño XP sigue rigurosamente el principio MS (mantenlo sencillo). Un diseño sencillo siempre se prefiere sobre una representación más compleja. Además, el diseño guía la implementación de una historia conforme se escribe: nada más y nada menos. Se desalienta el diseño de funcionalidad adicional porque el desarrollador supone que se requerirá después. Los conceptos más importantes de diseño en esta metodología son los siguientes:
(S. Pressman, 2010)

- Simplicidad, Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. (S. Pressman, 2010)

- Soluciones “Spike”, Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados “Spike”), para explorar diferentes soluciones. (S. Pressman, 2010)

- Recodificación (“Refactoring”), Consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de crearlo más simple, conciso y entendible. Las metodologías de XP sugieren re codificar cada vez que sea necesario. (S. Pressman, 2010)

- Metáforas, XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, así como guiar la estructura del mismo. Una buena metáfora debe ser fácil de comprender para el cliente y a su vez debe tener suficiente contenido como para que sirva de guía a la arquitectura del proyecto. (S. Pressman, 2010)

3.3.3 Codificación

Un concepto clave durante la actividad de codificación (y uno de los aspectos del que más se habla en la XP) es la programación por parejas. XP recomienda que dos personas trabajen juntas en una estación de trabajo con el objeto de crear código para una historia. Esto da un mecanismo para la solución de problemas en tiempo real (es frecuente que dos cabezas

piensen más que una) y para el aseguramiento de la calidad también en tiempo real (el código se revisa conforme se crea). (Joskowicz, 2008)

3.3.4 Pruebas

- Pruebas Unitarias, Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Test-Driven Programming”). Que todo código liberado pase correctamente las pruebas unitarias, es lo que habilita que funcione la propiedad colectiva del código. (Joskowicz, 2008)

- Detección y Corrección de Errores, Cuando se encuentra un error (“Bug”), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto. (Joskowicz, 2008)

- Pruebas de Aceptación, Son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El Cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta que pase correctamente todas las pruebas de aceptación. (Joskowicz, 2008)

CAPÍTULO IV. DESARROLLO DE LA TESIS

En la iteración 0 se lista los nombres de todas las historias de usuario para posteriormente ver sus descripciones completas en la iteración 1, donde son descritas a detalle.

4.1 Lista Maestra

En la tabla 3 se muestra la lista maestra de las historias de usuario para realizar la planeación sobre el proyecto.

Tabla 2
Lista Maestra de Historias de Usuarios.

NOMBRE DE HISTORIA	PRIORIDAD	ITERACIÓN
Interfaz Acceso y registro a la plataforma	Alta	1
Interfaz Visualización de productos turísticos	Media	1
Interfaz Registro de productos turísticos	Media	1
Interfaz Actualización de productos turísticos	Media	1
Visualización de productos	Alta	2
Búsquedas de recomendaciones de productos turísticos	Alta	2
Productos turísticos	Alta	2
Evaluación productos turísticos recomendaciones	alta	2

4.2 Fase 1: Planeación

4.2.2 Historias de Usuario

Para las siguientes historias de usuario se realizó el levantamiento de requerimientos al cliente, quien nos proporcionó la siguiente información que se muestra a continuación:

Tabla 3
Historia de Usuario 1. Acceso y Registro a la plataforma

Historia de usuario	
Numero: 1	Usuario: Administrador
Nombre de la historia: Acceso y registro a la plataforma	
Prioridad: alta	Riesgo de desarrollo: alto
Puntos estimados: 2	Iteración Asignada:1
Programador Responsable: Maricela	
Descripción: Para acceder a la plataforma se llevará a cabo el registro del usuario, el cual contará con un nombre de usuario y contraseña, posteriormente el administrador asignará los privilegios que cada usuario tendrá. Existirán los siguientes tipos de usuario: ● Administrador ● Turista ● Prestador	
Observaciones:	

Tabla 4
Historia de Usuario 2. Registro de productos turísticos

Historia de usuario	
Numero: 2	Usuario: Administrador
Nombre de la historia: Registro de productos turísticos	
Prioridad: media	Riesgo de desarrollo: alto
Puntos estimados: 3	Iteración Asignada: 1
Programador Responsable: Zuriel	
Descripción: interfaz para el registro de productos turísticos	

Observaciones:	

Tabla 5
Historia de Usuario 3. Visualización de productos turísticos

Historia de usuario	
Numero: 3	Usuario: Administrador
Nombre de la historia: Visualización de productos turísticos	
Prioridad: media	Riesgo de desarrollo: alto
Puntos estimados: 3	Iteración Asignada: 1
Programador Responsable: Zuriel	
Descripción: interfaz para la visualización de productos turísticos	
Observaciones:	

Tabla 6
Historia de Usuario 4. Actualización de productos turísticos

Historia de usuario	
Numero: 4	Usuario: Administrador
Nombre de la historia: Actualización de productos turísticos	
Prioridad: media	Riesgo de desarrollo: alto
Puntos estimados: 2	Iteración Asignada: 1
Programador Responsable: Zuriel y Elizabeth	
Descripción: interfaz para la actualización de productos turísticos	
Observaciones:	

Tabla 7
Historia de Usuario 5. Eliminación de productos turísticos

Historia de usuario	
Numero: 5	Usuario: Administrador
Nombre de la historia: Eliminación de productos turísticos	
Prioridad: media	Riesgo de desarrollo: alto
Puntos estimados: 3	Iteración Asignada: 1
Programador Responsable: Zuriel y Elizabeth	
Descripción: interfaz para la eliminación de productos turísticos	
Observaciones:	

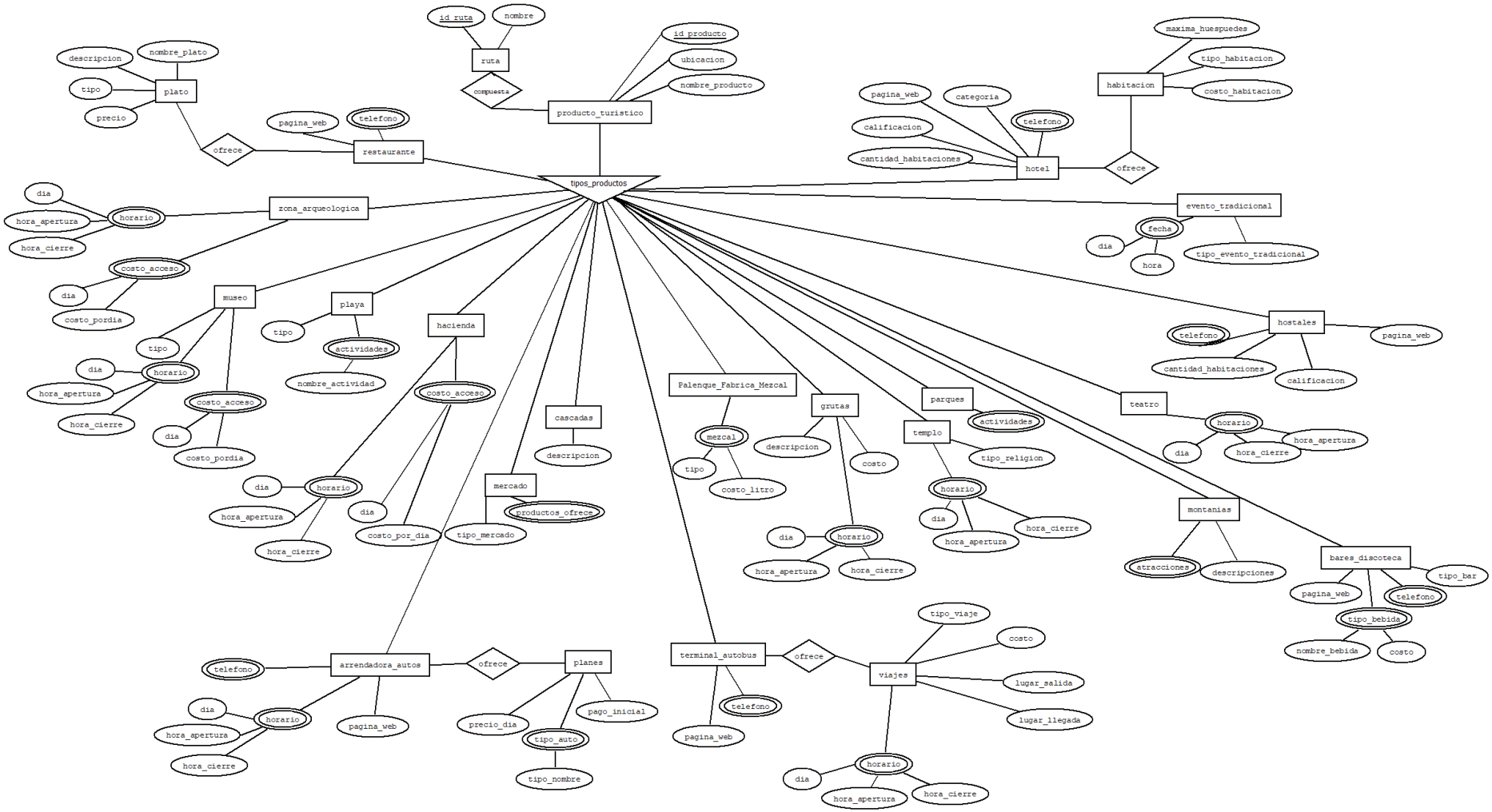
Tabla 8
Historia de Usuario 6. Visualización de productos turísticos

Historia de usuario	
Numero: 6	Usuario: Administrador
Nombre de la historia: Visualización de productos turísticos	
Prioridad: media	Riesgo de desarrollo: alto
Puntos estimados: 3	Iteración Asignada: 1
Programador Responsable: Zuriel	
Descripción: interfaz para la visualización de productos turísticos	
Observaciones:	

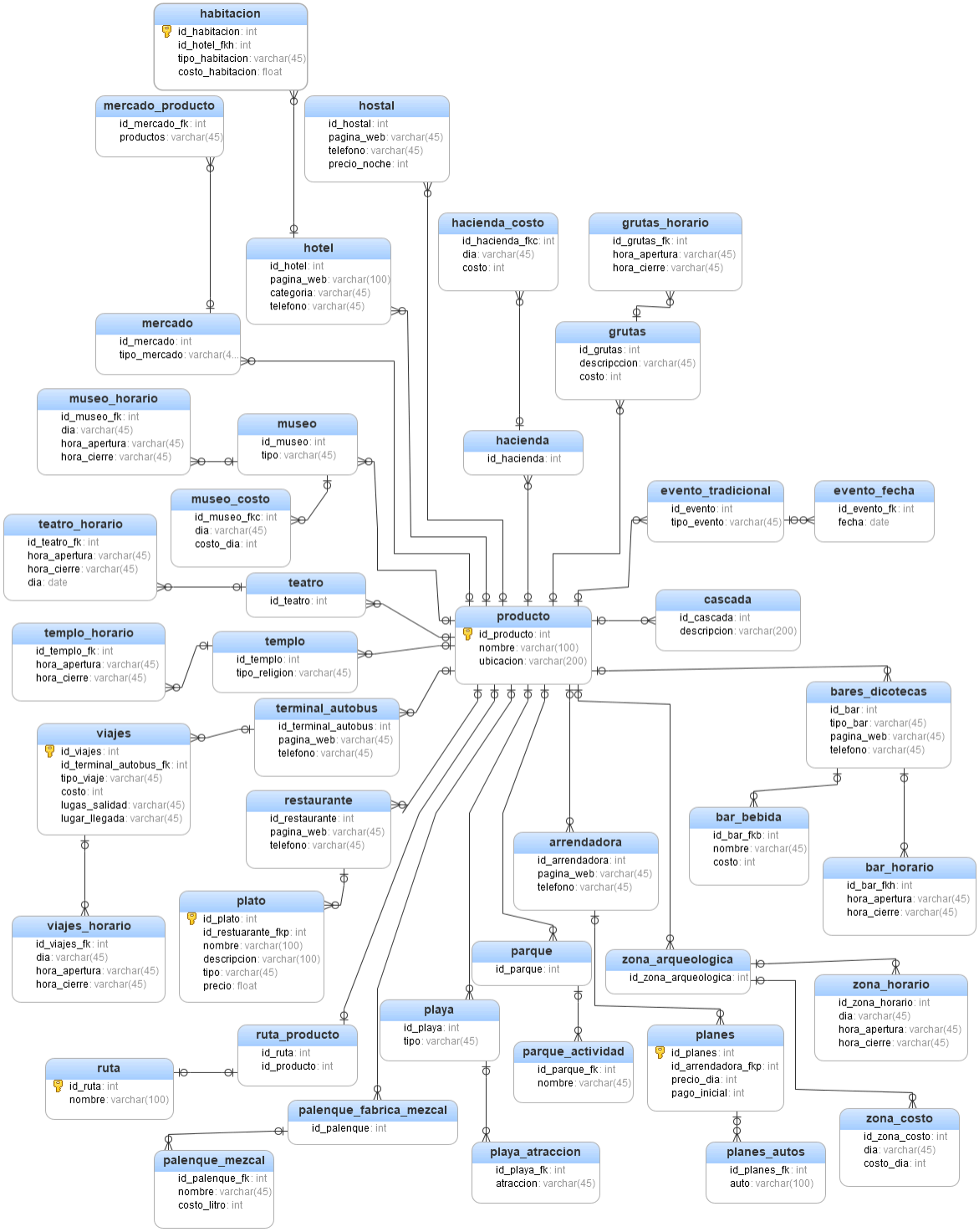
Para la fase de planeación de esta primera iteración se realizaron reuniones con el cliente para poder levantar los requerimientos del sistema. Se hicieron anotaciones sobre los módulos a desarrollar (ver en Anexos).

4.3 Fase 2: Diseño

4.3.1 Modelo E-R



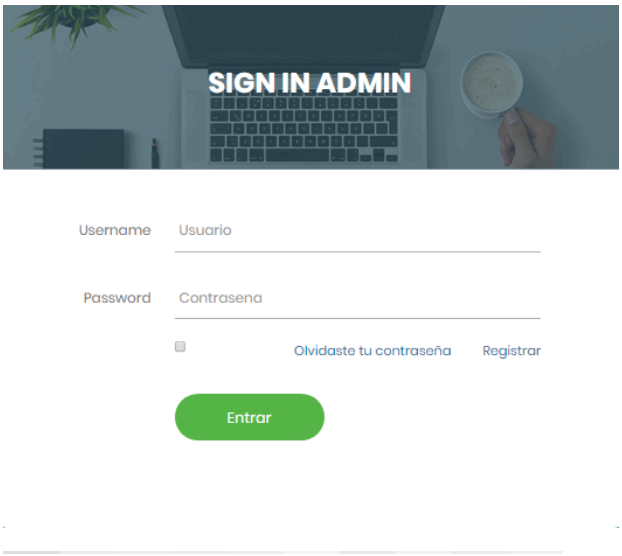
4.3.2 Diagrama relacional



4.3.3 Desarrollo de Interfaces

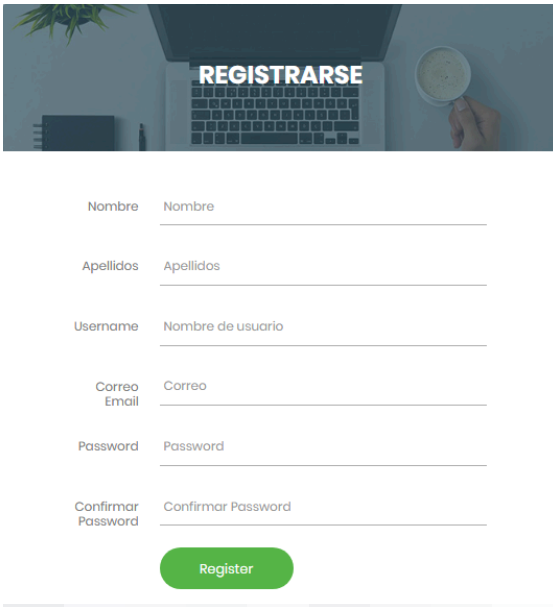
4.2.3.1 Interfaz inicio de sesión

Al iniciar el sistema se tendrá una interfaz donde el usuario podrá iniciar sesión, donde ingresa su nombre de usuario y contraseña.



4.3.1.2 Interfaz registro

El usuario podrá solicitar su registro al sistema mediante un formulario que podrá llenar con sus datos.



4.3.1.3 Interfaz registro de productos turísticos

En este apartado se puede agregar alguno de los productos turísticos donde se mostrará el formulario para ingresar los datos (aplica para todos los productos turísticos).

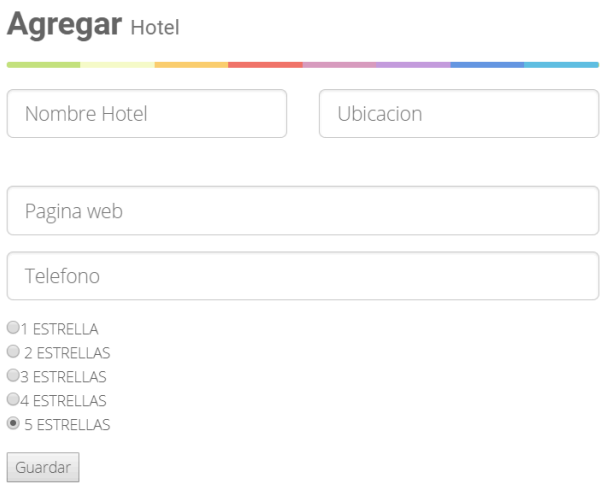


Ilustración 8 interfaz registro producto

4.3.1.4 Interfaz visualización productos turísticos

En la pantalla se listará en forma de tabla los productos turísticos que haya seleccionado anteriormente.

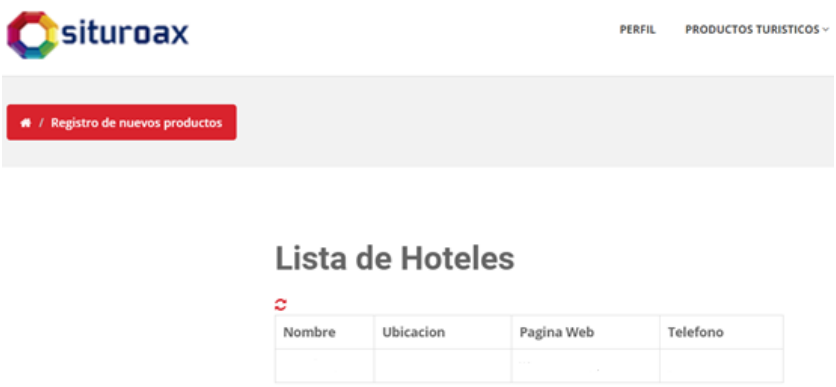


Ilustración 9 interfaz visualización de productos turísticos

4.3.1.5 Interfaz actualización productos turísticos

Al ingresar en la opción de modificar productos turísticos se mostrará una pantalla donde se listan los productos de ese tipo, al dar clic en el nombre de alguno de ellos se cargará un formulario con los datos para que pueda modificar algún campo en caso de error.

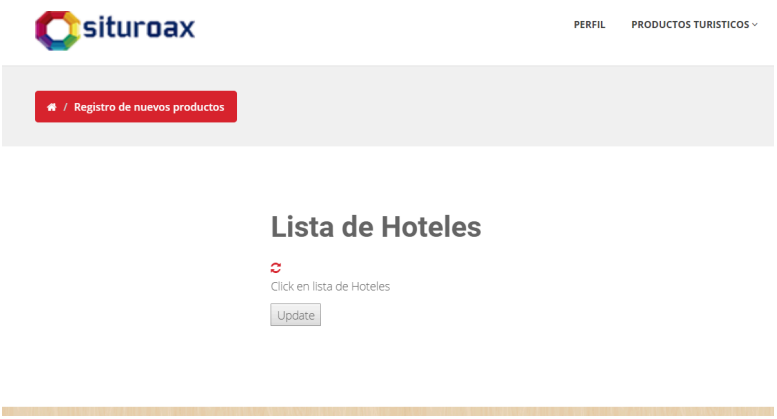


Ilustración 10 interface actualización

4.3.1.6 Interfaz eliminación productos turísticos

En la opción de eliminar producto turístico se mostrará una pantalla que aparecerá un listado del nombre del producto, al dar clic en alguno de ellos se mostraran sus datos o en caso contrario al dar clic al botón eliminar se borrara.

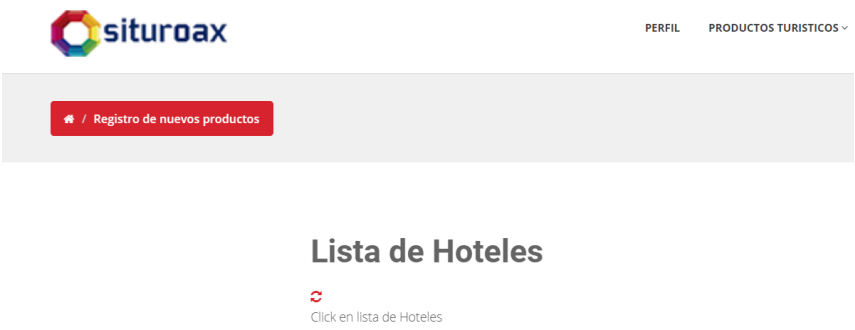


Ilustración 11 interface eliminación

4.4 Fase 3: Codificación

4.4.2 Codificación Modelo Vista Controlador

4.4.2.1 Controlador

La clase controlador hotel responde a los eventos del usuario e invoca peticiones al modelo cuando se hace una solicitud sobre información, es el intermediario entre los modelos y las vistas. (Ver código completo en anexos).

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Hotel extends CI_Controller{

        function __construct(){
                parent::__construct();
                // load Pagination library
    $this->load->library('pagination');

    // load URL helper
    $this->load->helper('url');
                $this->load->helper(array('form', 'url'));
                $this->load->library('form_validation');
                $this->load->library('security');
                $this->load->library('tank_auth');
                $this->lang->load('tank_auth');

                $this->load->model('usersModel');
                $this->load->model('hotelModel');
```

```
        }

function show_hotel_id() {
$id = $this->uri->segment(3);
$data['hoteles'] = $this->hotelModel->show_hoteles();
$data['single_hotel'] = $this->hotelModel->show_hotel_id($id);
$this->load->view('hoteles/editar_hotel', $data);
}

function show_hotel_id_d() {
$id = $this->uri->segment(3);
$data['hoteles'] = $this->hotelModel->show_hoteles();
$data['single_hotel'] = $this->hotelModel->show_hotel_id_d($id);
$this->load->view('hoteles/eliminar_hotel', $data);
}
function update_hotel() {
$id= $this->input->post('did');
$data = array(
'nombre'            => $this->input->post('nombre'),
'ubicacion'     => $this->input->post('ubicacion'),
'categoria'     => $this->input->post('categoria'),
'pagina_web'  => $this->input->post('pagina_web'),
'telefono'          => $this->input->post('telefono'),

);
```

4.3.2.2 Modelo

La clase modelo es la representación de la información, por lo tanto gestiona todos los accesos a dicha información, tanto como consultas y o actualizaciones, envía a las vistas parte de la información que en cada momento se solicita, para que sea mostrada al usuario. (ver código completo en anexos).

```
function verTodoHoteles($limite = 0){
   $this->db->limit($limit, $start);
   $query = $this->db->select('*')
            ->from('producto')
            ->join('hotel', 'id_producto = id_hotel')
            ->join('habitacion', 'id_hotel= id_hotel_fkh','LEFT')
            ->get();

    if ($query->num_rows() > 0){
    foreach ($query->result() as $row)
```

```
        {
            $data[] = $row;
        }

        return $data;
    }else{
     return FALSE;
    }
  }

function show_hoteles(){
  $this->db->select('*');
  $this->db->from('producto');
  $this->db->join('hotel', 'id_producto = id_hotel','INNER');
  $query = $this->db->get();
  $query_result = $query->result();
  return $query_result;
}

function show_hotel_id($data){

  $this->db->select('*');
  $this->db->from('producto');
  $this->db->join('hotel', 'id_producto = id_hotel','INNER');

  $this->db->where('id_hotel', $data);
  $query = $this->db->get();
  $result = $query->result();
  return $result;
}
```

4.3.2.3 Vista

Es la forma en que el modelo (información y lógica del negocio) es presentado al usuario de una forma sencilla, y comprensible. (Ver código completo en anexos).

```
<?php  $this->load->view('inicio/cabecera');?>
<?php  $this->load->view('inicio/body');?>

<?php

$nombre = array(
  'name'  => 'nombre',
```

```
  'id'  => 'nombre',
  'value' => set_value('nombre'),
  'maxlength' => 50,
  'size'  => 50,
  'class' => 'form-control input-lg',
  'placeholder' => 'Nombre Hotel'

);
$ubicacion = array(
  'name'  => 'ubicacion',
  'id'  => 'ubicacion',
  'value' => set_value('ubicacion'),
  'class' => 'form-control input-lg',
  'placeholder' => 'Ubicacion',
  'size'  => 50
);

$paginaweb = array(
  'name'  => 'paginaweb',
  'id'  => 'paginaweb',
  'value' => set_value('paginaweb'),
  'class' => 'form-control input-lg',
  'maxlength' => 100,
  'size'  => 50,
  'placeholder' => 'Pagina web'

);
$categoria = array(
  'name'  => 'categoria',
  'id'  => 'categoria',
  'value' => set_value('categoria'),
  'maxlength' => 80,
  'class' => 'form-control input-lg',
  'size'  => 50,

);
$telefono = array(
  'name'  => 'telefono',
  'id'  => 'telefono',
  'value' => set_value('telefono'),
  'maxlength' => 80,
  'size'  => 50,
  'class' => 'form-control input-lg',
  'placeholder' => 'Telefono',
);
```

```
$submit = array(
  'name' => 'submit',
  'id' => 'submit',
  'value' => 'Guardar',
  'title' => 'Guardar'
);


?>



<!-- copy-->
<!-- end header -->
<section id="inner-headline">
  <div class="container">
    <div class="row">
      <div class="col-lg-12">
        <ul class="breadcrumb">
          <li><a href="<?=base_url()?>index.php/Welcome/index"><i class="fa
fa-home"></i></a><i class="icon-angle-right"></i></li>

          <li class="active">Registro de nuevos productos</li>
        </ul>
      </div>
    </div>
  </div>
</section>
<!-- end copy>-->
<section id="content">
  <div class="container">

    <div class="row">
      <div class="col-xs-12 col-sm-8 col-md-6 col-sm-offset-2 col-md-offset-3">
       <form action="http://localhost/residencia/index.php/Hotel/crearHotel" method="post"
accept-charset="utf-8">
        <h2>Agregar <small>Hotel</small></h2>
        <hr class="colorgraph">
        <div class="row">
          <div class="col-xs-12 col-sm-6 col-md-6">
            <div class="form-group">
             <?php echo form_input($nombre); ?>
           </div>
         </div>
         <div class="col-xs-12 col-sm-6 col-md-6">
          <div class="form-group">
```

```
        <?php echo form_input($ubicacion); ?>
      </div>
     </div>
    </div>
    <div class="form-group">
     <?php echo form_input($paginaweb); ?>
   </div>
   <div class="form-group">
     <?php echo form_input($telefono); ?>
   </div>
   <div class="row">
     <div class="col-xs-12 col-sm-6 col-md-6">
       <div class="form-group">
        <div>
         <div class="form-check">
           <input type="radio" name="categoria" value="1 ESTRELLA" <?php echo
set_radio('categoria', '1 ESTRELLA', TRUE); ?> />1 ESTRELLA
         </div>

         <div class="form-check">
          <input type="radio" name="categoria" value="2 ESTRELLAS" <?php echo
set_radio('categoria', '2 ESTRELLAS', TRUE); ?> /> 2 ESTRELLAS
        </div>
        <div class="form-check">
         <input type="radio" name="categoria" value="3 ESTRELLAS" <?php echo
set_radio('categoria', '3 ESTRELLAS', TRUE); ?> />3 ESTRELLAS
       </div>
       <div class="form-check">
         <input type="radio" name="categoria" value="4 ESTRELLAS" <?php echo
set_radio('categoria', '4 ESTRELLAS', TRUE); ?> />4 ESTRELLAS
       </div>
       <div class="form-check">
        <input type="radio" name="categoria" value="5 ESTRELLAS" <?php echo
set_radio('categoria', '5 ESTRELLAS', TRUE); ?> /> 5 ESTRELLAS
      </div>

    </div>
  </div>

  <?php echo form_submit($submit);?>

  <?php echo form_close(); ?>
 </div>
</div>
</form>
</div>
```

```
</div>
</div>
</section>
<?php  $this->load->view('inicio/final');?>
```

4.5 Fase 4: Pruebas

4.5.1 Ingreso al sistema

El usuario si ya se encuentra registrado podrá ingresar al sistema utilizando su usuario y contraseña (password), si los datos son correctos se le mostrará la pantalla principal de carga del sistema, dependiendo del tipo de usuario si es administrador podrá tener acceso a las opciones de crear, visualizar, actualizar y eliminar de los productos turísticos, por otra parte, si es del tipo turista se mostrará la pantalla para buscar y evaluar los productos.

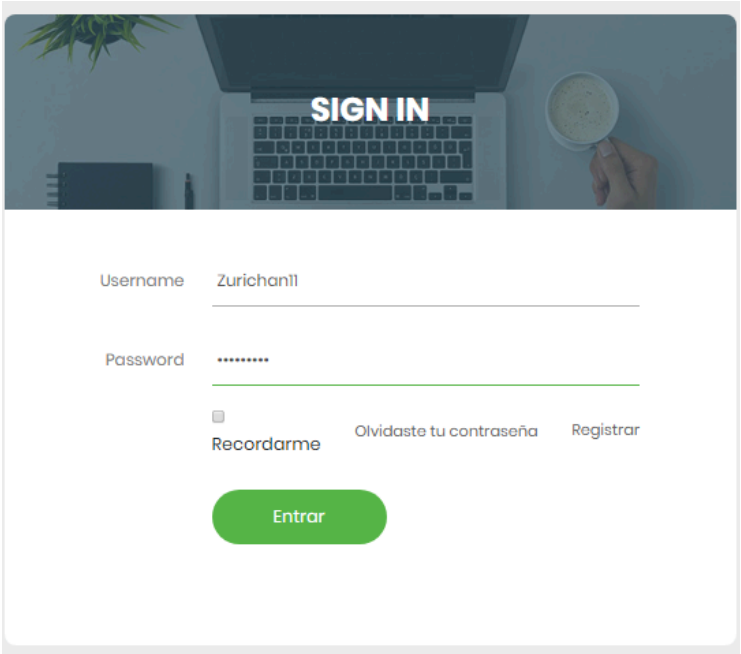


Ilustración 12 prueba inicio sesión

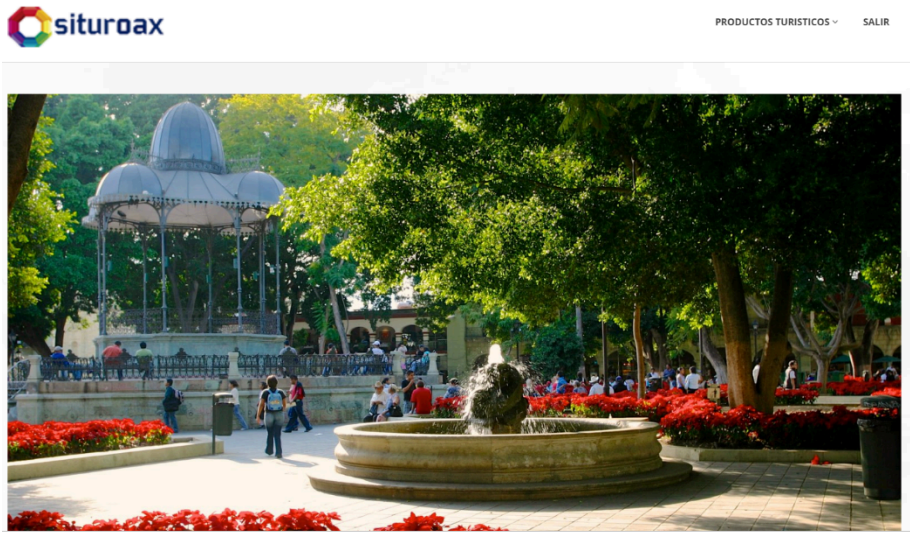


Ilustración 13 pantalla administrador

4.5.2 Registro de nuevos productos

Para el registro se deberán llenar todos los campos, en caso de no ser así y presionar el botón guardar, no lo hará y mostrará de nuevo la pantalla, sí los datos son correctos y todos los campos están completos se procederá a guardar la información.

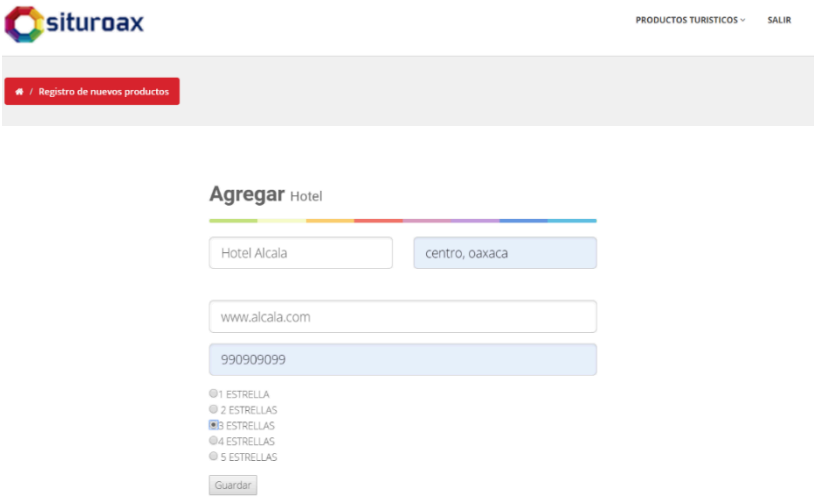


Ilustración 14 prueba registro producto

4.5.3 Visualización de los productos registrados

El usuario podrá seleccionar que tipo de producto desea ver, sí elige un producto en el cual no hay registros guardados no mostrará nada, si hay elementos guardados los podrá observar en forma de tabla, a partir de más de 7 elementos mostrará una paginación.

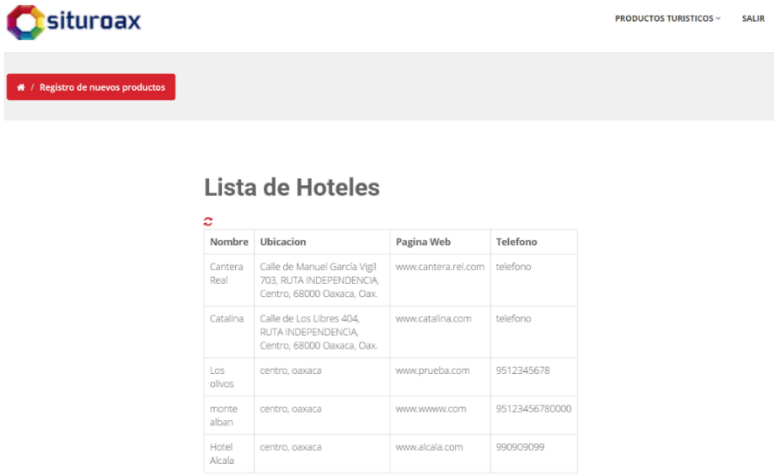


Ilustración 15 prueba visualización producto

4.5.3 Actualización de productos turísticos

El usuario dará clic en la pestaña de actualizar, si no hay ningún producto registrado no aparecerá nada, si hay al menos uno, se mostrara una lista con los nombres al dar clic sobre uno de ellos donde podrá ver un formulario con los datos de ese producto, para saber cuál actualizar posteriormente, dará clic en el botón update, si los datos son correctos se actualizara en caso contrario no dejara actualizar.

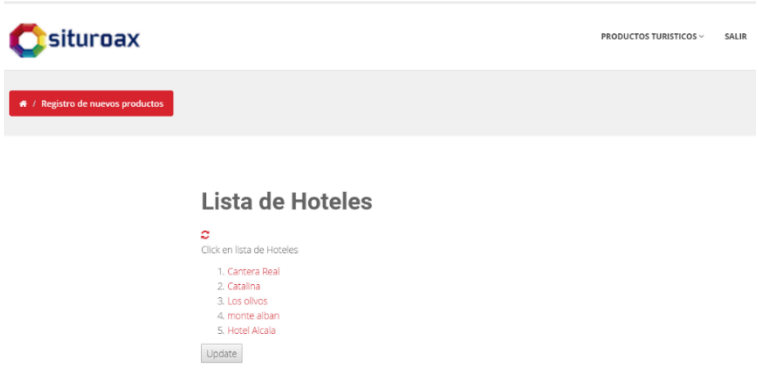


Ilustración 16 prueba listado actualización

Ilustración 17 prueba actualización

4.5.4 Eliminación de productos turísticos

El usuario al dar clic en la pestaña de eliminar y no hay ningún producto registrado no mostrara nada, en caso contrario se mostrará una lista, al seleccionar alguno de estos mostrara sus datos y un botón para eliminarlo, al dar clic en el botón se eliminará y actualizara la página.

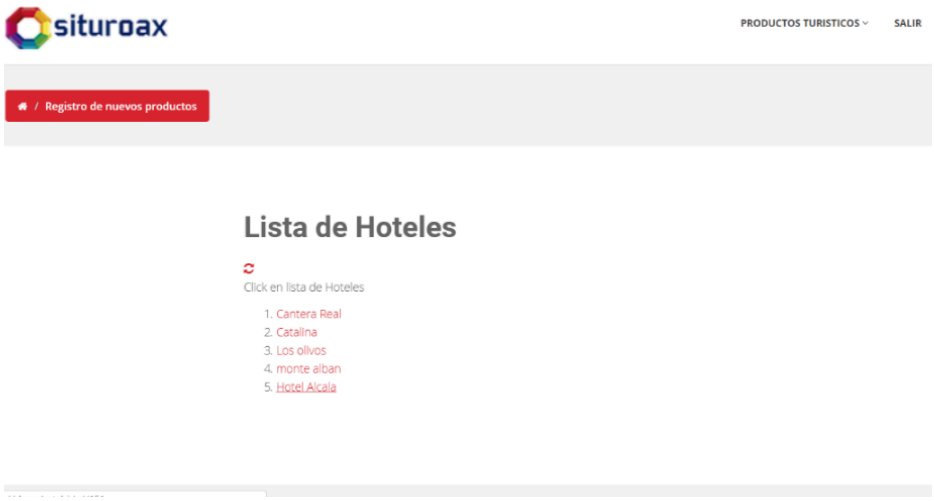


Ilustración 18 prueba listado eliminar

En la ilustración 17 se puede ver la prueba realizada sobre la eliminación de un producto.

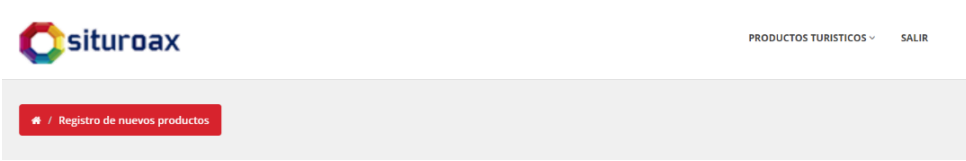


Ilustración 19 prueba información a eliminar

Al igual que la ilustración 18, vuelve a mostrar en la pantalla la lista de en este caso Hoteles.

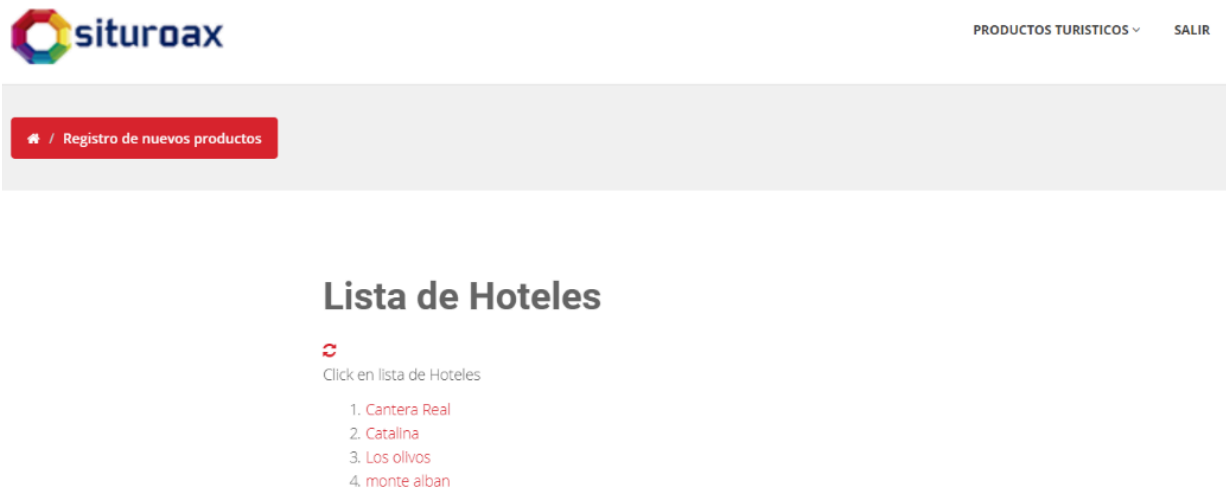


Ilustración 20 listado eliminación actualizado

4.6 Fase 1: Planeación

4.6.1 Historias de Usuario

En la segunda iteración se muestra lo que el cliente nos proporcionó para la realización de tal proyecto.

Tabla 8
Historia de Usuario 6. Visualización de productos turísticos

Historia de usuario	
Numero: 6	Usuario: Administrador
Nombre de la historia: Visualización de Productos Turísticos	
Prioridad: alta	Riesgo de desarrollo: bajo
Puntos estimados: 2	Iteración Asignada: 2
Programador Responsable: Maricela	
Descripción: El Administrador captura los productos para poder visualizarlos en la interfaz	
Observaciones:	

Tabla 9.
Historia de Usuario 7. Búsqueda de recomendación de productos turísticos

Historia de usuario	
Numero: 7	Usuario: Turista
Nombre de la historia: Búsqueda de Recomendación de Productos Turísticos	
Prioridad: alta	Riesgo de desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 2
Programador Responsable: Maricela	
Descripción: Interfaz que permite al usuario realizar las búsquedas de recomendaciones de productos turísticos según su perfil (Hoteles, Restaurantes)	
Observaciones:	

Tabla 10
Historia de Usuario 8. Resultados de recomendación de productos turísticos

Historia de usuario	
Numero: 8	Usuario: Turista
Nombre de la historia: Resultados de Recomendación de Productos Turísticos	
Prioridad: alta	Riesgo de desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 2
Programador Responsable: Maricela	
Descripción: Interfaz para que el turista pueda visualizar las recomendaciones de productos turísticos (Hoteles y Restaurantes)	
Observaciones:	

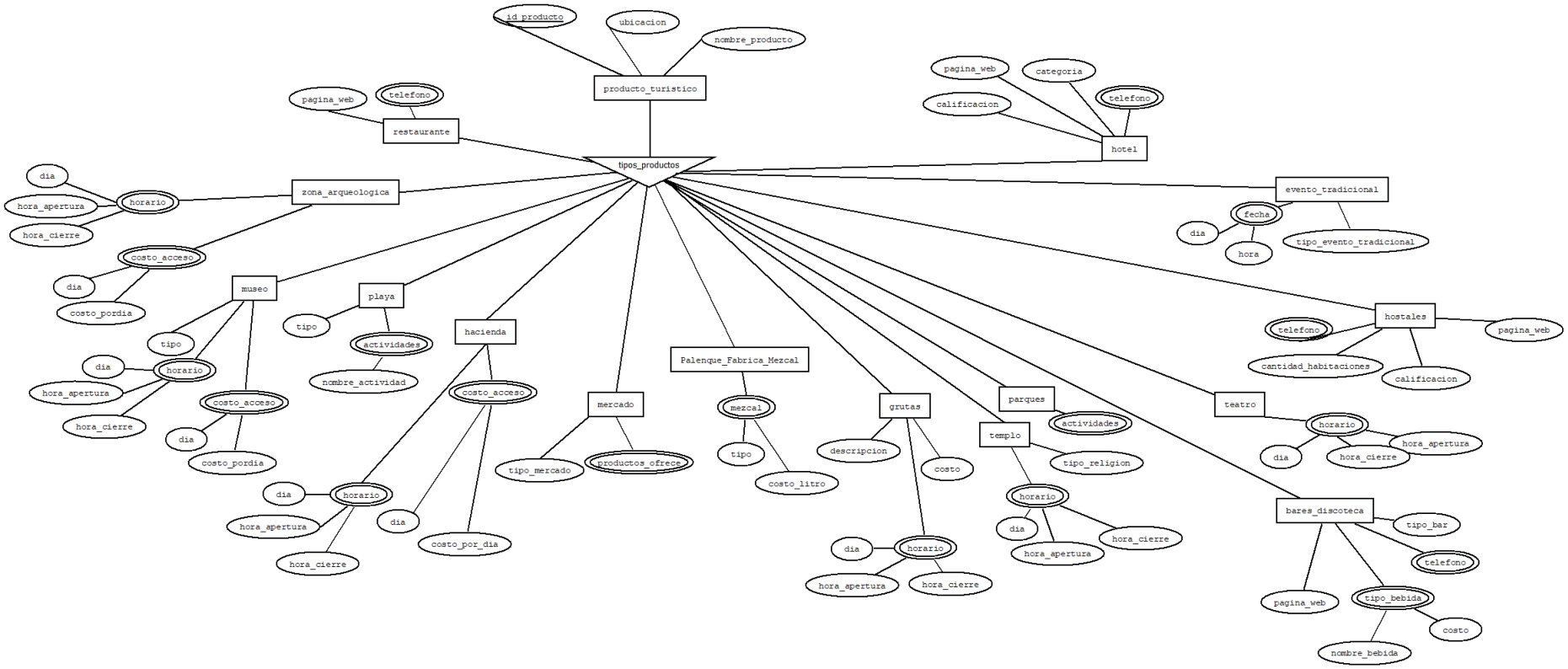
Tabla 11
Historia de Usuario 9. Evaluación de productos turísticos recomendados

Historia de usuario	
Numero: 9	Usuario: Turista
Nombre de la historia: Evaluación de Productos Turísticos Recomendados	
Prioridad: alta	Riesgo de desarrollo: Alto
Puntos estimados: 2	Iteración Asignada: 2
Programador Responsable: Maricela	
Descripción: Interfaz que permite al usuario evaluar los productos que se le recomendaron (Hoteles y Restaurantes)	
Observaciones:	

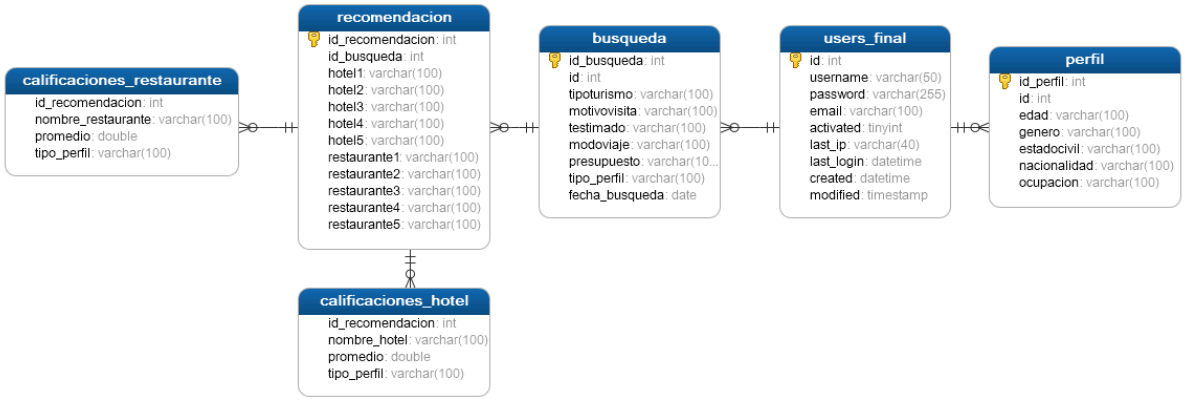
Dentro de esta fase de planeación, se realizaron reuniones con el cliente para definir los requerimientos de los módulos a desarrollar. Se anexan al final del documento.

4.7 Fase 2: Diseño

4.7.1 Diagrama E-R



4.7.2 Diagrama relacional



Para esta fase se realizan los diseños de las interfaces.

4.7.3 Interfaz de Búsqueda de Recomendación de Productos Turísticos

La pantalla o interfaz de inicio que visualiza el usuario al iniciar sesión en el sistema.

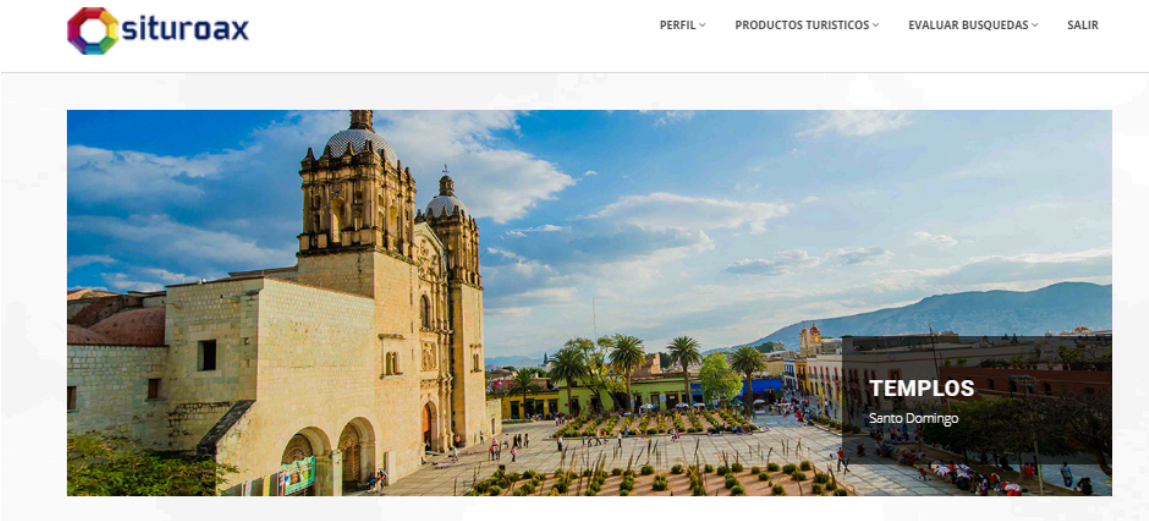


Ilustración 21 Pantalla de inicio búsquedas

La ilustracion 20 muestra el formulario que el usuario podrá observar para realizar la busqueda de los servicios turisticos, despues de haber seleccionado su tipo de turismo, el motivo de visita, modo de viaje, tiempo estimado y su presupuesto.

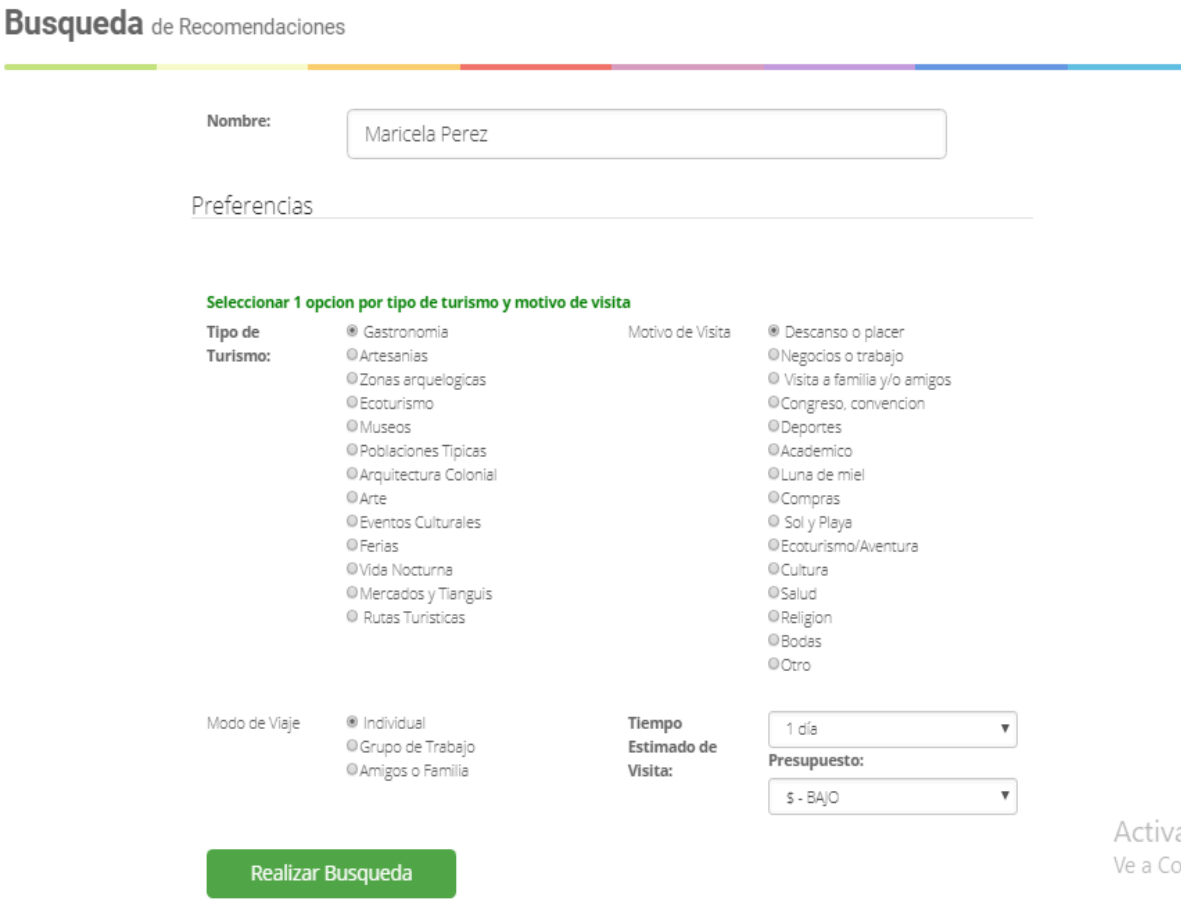


Ilustración 22 Formulario de búsqueda

4.7.4 Interfaz de Resultados de Recomendación de Productos Turísticos (Hoteles y Restaurantes)

En la interfaz de resultado de la búsqueda el usuario lograra ver un mapa del lado izquierdo donde punteé las sugerencias de los servicios turísticos, donde en el lado izquierdo se encuentran las sugerencias.

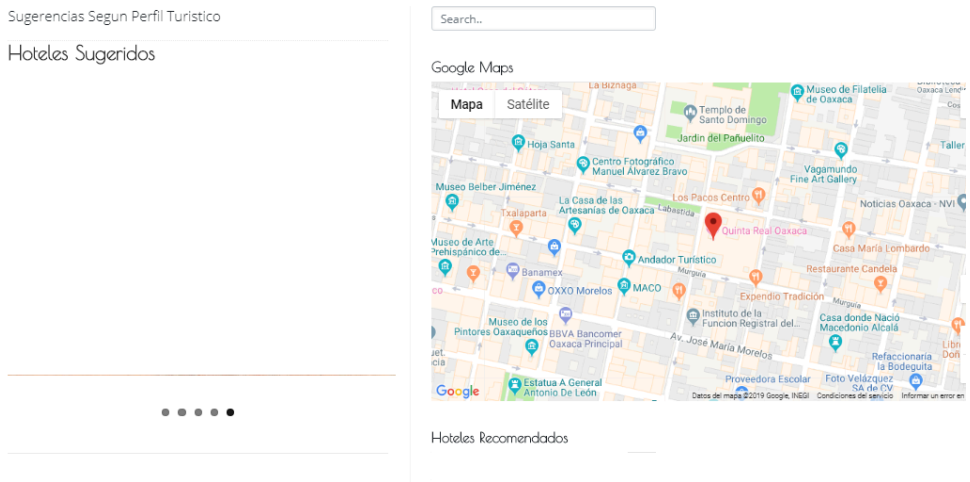


Ilustración 23 interfaz de resultados de búsqueda

4.7.5 Interfaz de Evaluación de Recomendación de Productos Turísticos

En las interfaces de evaluación se hace lo mismo para hoteles y restaurantes, en este apartado los usuarios turistas podrán realizar una breve evaluación para calificar los servicios que visitó.

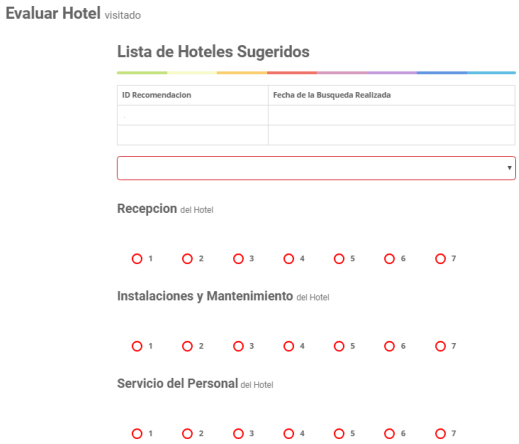


Ilustración 24 Interfaz de evaluación

4.8 Fase 3: Codificación

4.8.1 Módulo de Búsqueda de Recomendación de Productos Turísticos

(View)

```
      <form action="<?php echo base_url()?>index.php/turista/realizarbusqueda"
method="post" accept-charset="utf-8">

           <div class="form-group row">
          <label  class="col-sm-2 col-form-label">Nombre:</label>
          <div class="col-sm-9">
         <input type="text"  size="50" name="nombre" class="form-control
input-lg" id="nombre" value="<?php echo $usuarios['firstname'] ?> <?php echo
$usuarios['lastname']; ?>">
       </div>

      </div>

      <fieldset class="form-group">
        <div class="row">
          <legend class="col-form-label col-sm-15 pt-0">Preferencias </legend>
        </div>
      </fieldset>
      <label style="color: green;"> Seleccionar 1 opcion por tipo de turismo y
motivo de visita</label>
      <br>

      <div class="form-group row">
        <label  class="col-sm-2 col-form-label">Tipo de Turismo:</label>
        <div class="col-sm-4">
          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Gastronomia" <?php
echo  set_radio('tipoturismo', 'Gastronomia', TRUE); ?> /> Gastronomia
          </div>

          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Artesanias" <?php
echo  set_radio('tipoturismo', 'Artesanias', FALSE); ?> />Artesanias
          </div>
          <div class="form-check">
           <input type="radio" name="tipoturismo" value="Zonas arquelogicas"
<?php echo  set_radio('tipoturismo', 'Zonas arquelogicas', FALSE); ?> />Zonas
arquelogicas
          </div>
          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Ecoturismo" <?php
echo  set_radio('tipoturismo', 'Ecoturismo', FALSE); ?> />Ecoturismo
```

```
          </div>
          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Museos" <?php echo
set_radio('tipoturismo', 'Museos', FALSE); ?> />Museos
          </div>
          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Poblaciones Tipicas"
<?php echo  set_radio('tipoturismo', 'Poblaciones Tipicas', FALSE); ?>
/>Poblaciones Tipicas
          </div>
          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Arquitectura
Colonial" <?php echo  set_radio('tipoturismo', 'Arquitectura Colonial', FALSE);
?> />Arquitectura Colonial
          </div>
          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Arte" <?php echo
set_radio('tipoturismo', 'Arte', FALSE); ?> />Arte
          </div>
          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Eventos Culturales"
<?php echo  set_radio('tipoturismo', 'Eventos Culturales', FALSE); ?> />Eventos
Culturales
          </div>
          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Ferias" <?php echo
set_radio('tipoturismo', 'Ferias', FALSE); ?> />Ferias
          </div>
          <div class="form-check">
            <input type="radio" name="tipoturismo" value="Vida Nocturna" <?php
echo  set_radio('tipoturismo', 'Vida Nocturna', FALSE); ?> />Vida Nocturna
          </div>
          <div class="form-check">

            <input type="radio" name="tipoturismo" value="Mercados y Tianguis"
<?php echo  set_radio('tipoturismo', 'Mercados y Tianguis', FALSE); ?>
/>Mercados y Tianguis
          </div>
          <div class="form-check">
           <input type="radio" name="tipoturismo" value="Rutas Turisticas"
<?php echo  set_radio('tipoturismo', 'Rutas Turisticas', FALSE); ?> /> Rutas
Turisticas
          </div>
        </div>
```

```
        <div class="col-sm-2">Motivo de Visita</div>
        <div class="col-sm-4">
          <div class="form-check">
           <input type="radio" name="motivovisita" value="Descanso o placer"
<?php echo  set_radio('motivovisita', 'Descanso o placer', TRUE); ?> />
Descanso o placer
         </div>
         <div class="form-check">
           <input type="radio" name="motivovisita" value="Negocios o trabajo"
<?php echo  set_radio('motivovisita', 'Negocios o trabajo', FALSE); ?>
/>Negocios o trabajo
         </div>
         <div class="form-check">
          <input type="radio" name="motivovisita" value="Visita a familia y/o
amigos" <?php echo  set_radio('motivovisita', 'Visita a familia y/o amigos',
FALSE); ?> /> Visita a familia y/o amigos
        </div>
        <div class="form-check">
          <input type="radio" name="motivovisita" value="Congreso-Convencion"
<?php echo  set_radio('motivovisita', 'Congreso, convencion', FALSE); ?>
/>Congreso, convencion
        </div>
        <div class="form-check">
          <input type="radio" name="motivovisita" value="Deportes" <?php echo
set_radio('motivovisita', 'Deportes', FALSE); ?> />Deportes
        </div>
        <div class="form-check">
          <input type="radio" name="motivovisita" value="Academico" <?php echo
set_radio('motivovisita', 'Academico', FALSE); ?> />Academico
        </div>
        <div class="form-check">
          <input type="radio" name="motivovisita" value="Luna de miel" <?php
echo  set_radio('motivovisita', 'Luna de miel', FALSE); ?> />Luna de miel
        </div>
        <div class="form-check">
          <input type="radio" name="motivovisita" value="Compras" <?php echo
set_radio('motivovisita', 'Compras', FALSE); ?> />Compras
        </div>
        <div class="form-check">
         <input type="radio" name="motivovisita" value="Sol y Playa" <?php echo
set_radio('motivovisita', 'Sol y Playa', FALSE); ?> /> Sol y Playa
       </div>
       <div class="form-check">
```

```
        <input type="radio" name="motivovisita" value="Ecoturismo/Aventura"
<?php echo  set_radio('motivovisita', 'Ecoturismo/Aventura', FALSE); ?>
/>Ecoturismo/Aventura
      </div>
      <div class="form-check">
        <input type="radio" name="motivovisita" value="Cultura" <?php echo
set_radio('motivovisita', 'Cultura', FALSE); ?> />Cultura
      </div>
      <div class="form-check">
        <input type="radio" name="motivovisita" value="Salud" <?php echo
set_radio('motivovisita', 'Salud', FALSE); ?> />Salud
      </div>
      <div class="form-check">
       <input type="radio" name="motivovisita" value="Religion" <?php echo
set_radio('motivovisita', 'Religion', FALSE); ?> />Religion
     </div>
     <div class="form-check">
       <input type="radio" name="motivovisita" value="Bodas" <?php echo
set_radio('motivovisita', 'Bodas', FALSE); ?> />Bodas
     </div>
     <div class="form-check">
       <input type="radio" name="motivovisita" value="Otro" <?php echo
set_radio('motivovisita', 'Otro', FALSE); ?> />Otro
     </div>
   </div>
  </div>
</div>

<div class="form-group row">

   <div class="col-sm-2">Modo de Viaje</div>
       <div class="col-sm-4">
         <div class="form-check">
          <input type="radio" name="modoviaje" value="Individual" <?php echo
set_radio('modoviaje', 'Individual', TRUE); ?> /> Individual
        </div>
        <div class="form-check">
          <input type="radio" name="modoviaje" value="Grupo de Trabajo" <?php
echo  set_radio('modoviaje', 'Grupo de Trabajo', FALSE); ?> />Grupo de Trabajo
        </div>
        <div class="form-check">
          <input type="radio" name="modoviaje" value="Amigos o Familia" <?php
echo  set_radio('modoviaje', 'Amigos o Familia', FALSE); ?> />Amigos o Familia
        </div>
      </div>
```

```
  <label  class="col-sm-2 col-form-label">Tiempo Estimado de Visita:</label>
  <div class="col-sm-4">
    <select class="chosen-select form-control" name="tiempoestimado">
      <option value="1 día" <?php echo  set_select('tiempoestimado', '1 día',
FALSE); ?> >1 día</option>
      <option value="2 días" <?php echo  set_select('tiempoestimado', '2
días'); ?> > 2 días</option>
      <option value="3 días" <?php echo  set_select('tiempoestimado', '3
días'); ?> > 3 días</option>
      <option value="4 días" <?php echo  set_select('tiempoestimado', '4
días'); ?> > 4 días</option>
      <option value="5 días" <?php echo  set_select('tiempoestimado', '5
días'); ?> > 5 días</option>
      <option value="6 días" <?php echo  set_select('tiempoestimado', '6
días'); ?> > 6 días</option>
      <option value="7 días" <?php echo  set_select('tiempoestimado', '7
días'); ?> > 7 días</option>
    </select>
  </div>

    <label  class="col-sm-2 col-form-label">Presupuesto:</label>
  <div class="col-sm-4">
    <select class="chosen-select form-control" name="presupuesto">
      <option value="$-BAJO" <?php echo  set_select('presupuesto', '$-BAJO',
FALSE); ?> >$ - BAJO</option>
      <option value="$$-MEDIO" <?php echo  set_select('presupuesto',
'$$-MEDIO'); ?> >$$ - MEDIO</option>
      <option value="$$$-ALTO" <?php echo  set_select('presupuesto',
'$$$-ALTO'); ?> > $$$ - ALTO</option>
     </select>
  </div>



</div>




<div class="form-group row">
 <div  class="col-sm-4">
   <?php echo form_submit($submit);?>
 </div>
 <?php echo form_close(); ?>
```

```
</div>
```

(Controllers)

```
	public function realizarbusqueda(){
		$this->load->library('form_validation');
		$this->load->model('turistaModel');
		$data['usernombre'] = $this->tank_auth->get_username();
		$result = $this->db->query('select * from users where id =
'.$this->tank_auth->get_user_id());
		$result = $result->result_array();
		$data['usuarios'] = $result[0];
		$id = $this->tank_auth->get_user_id();
		if ($this->input->post('submit')) {

			$prefturisticas = $this->input->post('tipoturismo');
			$motivovisita =$this->input->post('motivovisita');
			$tiempoestimado =$this->input->post('tiempoestimado');
			$modoviaje = $this->input->post('modoviaje');
			$presupuesto = $this->input->post('presupuesto');
			if (($motivovisita == 'Descanso o placer')|| ($motivovisita == 'Salud') ){
				$perfilturista = 'Turista Convencional';
			}elseif (($motivovisita == 'Sol y Playa')) {
				$perfilturista = 'Turista Sol y Playa';
			}elseif (($motivovisita == 'Cultural') || ($motivovisita == 'Bodas') ||
($prefturisticas == 'Mercados y Tianguis')|| ($prefturisticas == 'Arte') || ($prefturisticas ==
'Museos') || ($prefturisticas == 'Arquitectura Colonial') || ($prefturisticas == 'Zonas
arqueologicas') || ($prefturisticas == 'Artesanias') || ($prefturisticas == 'Rutas Turisticas')) {
				$perfilturista = 'Turista Cultural';
			}elseif ($motivovisita == 'Deportes') {
				$perfilturista = 'Turista Deportivo';
			}elseif (($motivovisita == 'Negocios') || ($motivovisita ==
'Congreso-Convencion')) {
				$perfilturista = 'Turista Negocios';
			}elseif (($motivovisita == 'Religion') || ($prefturisticas == 'Eventos
Culturales') || ($prefturisticas == 'Ferias') || ($prefturisticas == 'Gastronomia') || ($prefturisticas
== 'Poblaciones Tipicas') ||  ($prefturisticas == 'Compras')) {
				$perfilturista = 'Turista de Costumbres y Tradiciones';

			}elseif ($motivovisita == 'Academico') {
				$perfilturista = 'Turista Academico Cientifico';
			}elseif (($motivovisita == 'Ecoturismo/Aventura') || ($prefturisticas ==
'Ecoturismo')) {
				$perfilturista = 'Turista de Aventura Ecoturismo';
```

```
                        }elseif (($motivovisita == 'Luna de miel') || ($motivovisita == 'Visita a
familia y/o amigos') || ($prefturisticas == 'Vida Nocturna') || ($motivovisita == 'Otro')) {
                                $perfilturista = 'Turista Social';
                        }


                        $querybusqueda = "insert into busqueda
values('','$id','$prefturisticas','$motivovisita'
,'$tiempoestimado','$modoviaje','$presupuesto','$perfilturista')";
                        $this->db->query($querybusqueda);
                        $id_fk_busqueda = $this->db->insert_id();


                        exec("Rscript
C:/xampp/htdocs/Residencia2018/assets/ejecutaR/scriptHotel.R \"$perfilturista\"");
                        exec("Rscript
C:/xampp/htdocs/Residencia2018/assets/ejecutaR/scriptRestaurante.R \"$perfilturista\"");


                //
exec("C:/xampp/htdocs/Residencia2018/assets/ejecutaR/scriptRestaurante.R");
                        if (($handle =
fopen("C:/xampp/htdocs/Residencia2018/assets/ejecutaR/prueba1.csv", "r")) !== FALSE) {
                                $csvs = [];
                                while(! feof($handle)) {
                                        $csvs[] = fgetcsv($handle);
                                }
                                $datas = [];
                                $column_names = [];
                                foreach ($csvs[0] as $single_csv) {
                                        $column_names[] = $single_csv;
                                }
                                foreach ($csvs as $key => $csv) {
                                        if ($key === 0) {
                                                continue;
                                        }
                                        foreach ($column_names as $column_key =>
$column_name) {
                                                $datas[$key-1][$column_name] =
$csv[$column_key];
                                        }
                                }
                                $json = json_encode($datas);
                                        // $json = $json->array();
                                fclose($handle);
                        }
```

```
$jsonh = json_decode($json);


$query1 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = "'.$jsonh[0]->hotel.'"');
$row1 = $query1->row_array();
$query2 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = "'.$jsonh[1]->hotel.'"');
$row2 = $query2->row_array();
$query3 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = "'.$jsonh[2]->hotel.'"');
$row3 = $query3->row_array();
$query4 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = "'.$jsonh[3]->hotel.'"');
$row4 = $query4->row_array();
$query5 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = "'.$jsonh[4]->hotel.'"');
$row5 = $query5->row_array();

$hotel1 = $jsonh[0]->hotel;
$hotel2 = $jsonh[1]->hotel;
$hotel3 = $jsonh[2]->hotel;
$hotel4 = $jsonh[3]->hotel;
$hotel5 = $jsonh[4]->hotel;


$data['ubicacion1'] =  $row1['ubicacion'] ;
$data['ubicacion2'] =  $row2['ubicacion'] ;
$data['ubicacion3'] =  $row3['ubicacion'] ;
$data['ubicacion4'] =  $row4['ubicacion'] ;
$data['ubicacion5'] =  $row5['ubicacion'] ;
$data['json'] =  $json ;
$data['jsonh'] =  $jsonh ;

if (($handle =
fopen("C:/xampp/htdocs/Residencia2018/assets/ejecutaR/prueba2.csv", "r")) !== FALSE) {
        $csvs = [];
        while(! feof($handle)) {
                $csvs[] = fgetcsv($handle);
        }
        $datas = [];
        $column_names = [];
        foreach ($csvs[0] as $single_csv) {
                $column_names[] = $single_csv;
        }
        foreach ($csvs as $key => $csv) {
```

```
                                        if ($key === 0) {
                                                continue;
                                        }
                                        foreach ($column_names as $column_key =>
$column_name) {
                                                $datas[$key-1][$column_name] =
$csv[$column_key];
                                        }
                                }
                                $json2 = json_encode($datas);
                                        // $json = $json->array();
                                fclose($handle);
                        }

                        $jsonr = json_decode($json2);
                        $query6 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = '".$jsonr[0]->restaurante."');
                        $row6 = $query6->row_array();
                        $query7 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = '".$jsonr[1]->restaurante."');
                        $row7 = $query7->row_array();
                        $query8 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = '".$jsonr[2]->restaurante."');
                        $row8 = $query8->row_array();
                        $query9 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = '".$jsonr[3]->restaurante."');
                        $row9 = $query9->row_array();
                        $query10 = $this->db->query('SELECT ubicacion FROM producto
WHERE nombre = '".$jsonr[4]->restaurante."');
                        $row10 = $query10->row_array();

                        $restaurante1 = $jsonr[0]->restaurante;
                        $restaurante2 = $jsonr[1]->restaurante;
                        $restaurante3 = $jsonr[2]->restaurante;
                        $restaurante4 = $jsonr[3]->restaurante;
                        $restaurante5 = $jsonr[4]->restaurante;


                        date_default_timezone_set('America/Mexico_City');
                        $fecha = date("Y-m-d") ;


                        $queryrecomendacion = "insert into recomendacion
values('','$id_fk_busqueda','$hotel1','$hotel2','$hotel3','$hotel4','$hotel5','$restaurante1','$rest
aurante2','$restaurante3','$restaurante4','$restaurante5','$fecha')";
                        $this->db->query($queryrecomendacion);
```

```
                        $data['rubicacion1'] =  $row1['ubicacion'] ;
                        $data['rubicacion2'] =  $row2['ubicacion'] ;
                        $data['rubicacion3'] =  $row3['ubicacion'] ;
                        $data['rubicacion4'] =  $row4['ubicacion'] ;
                        $data['rubicacion5'] =  $row5['ubicacion'] ;
                        $data['json2'] =  $json2 ;
                        $data['jsonr'] =  $jsonr ;


                        $filename =
'C:/xampp/htdocs/Residencia2018/assets/ejecutaR/HotelesVisitados.csv';
                        $csvArray = $this->ImportCSV2Array($filename);

                        foreach ($csvArray as $row)
                        {
                                $resultado[] = $row['hoteles_visitados'];
                        }
                        $data['hoteles_visitados'] = $resultado;


                        $filename2 =
'C:/xampp/htdocs/Residencia2018/assets/ejecutaR/RestaurantesVisitados.csv';
                        $csvArray2 = $this->ImportCSV2Array($filename2);

                        foreach ($csvArray2 as $row)
                        {
                                $resultado2[] = $row['restaurantes_visitados'];
                        }
                        $data['restaurantes_visitados'] = $resultado2;

                        $this->load->view('busqueda/resultadobusqueda', $data);


                }


        }
```

(Models)

```
function guardarbusqueda($id,$tipoturismo,$motivovisita,$testimado,$modoviaje,$presupuesto)
  {
```

```
 $query="insert into busqueda values(','$id','$tipoturismo','$motivovisita'
,'$testimado','$modoviaje','$presupuesto')";

 $this->db->query($query);
 }
```

3.8.2 Código En R que ejecuta el Algoritmo de Recomendación

```
setwd("C:/xampp/htdocs/Residencia2018/assets/ejecutaR")
cat("\014") ## clear console
#file.remove("envs.txt") ## remove output file if exists
rm(list=ls()) ## Reset Environment

args <- commandArgs(TRUE)
tipo <-  args[1]
library(PerformanceAnalytics)

library(readxl)
DatosHotelR <-
read_excel("C:/xampp/htdocs/Residencia2018/assets/ejecutaR/DatosHotelR.xlsx",
   sheet = "Sheet 1")
colnames(DatosHotelR)<-c("Abu","Aitana","Angel Inn","Anua","Azucenas","Azul
Cielo","Aurora","Azul Oaxaca","Boutique Casa Cid de Leon","Boutique De la
Parra","Calenda","Camba","Cantera Real","Casa Antigua","Casa Bonita","Casa Carlota","Casa
Catrina","Casa Colon","Casa Colonial","Casa Conzatti","Casa Cue","Casa de Adobe","Casa de
las Flores","Casa de los Frailes","Casa de Siete Balcones","Casa del Sotano","Casa las
Mercedes","Casa los Cantaros","Casa Oaxaca","Casa Sierra Azul","Casa
Vertiz","Casantica","Casona Oaxaca","Catrina de Alcala","City Express","Dainzu","Del
Marquesado","El Principal","Esperanza","Fiesta Inn Oaxaca","Florida","Fortin Plaza","Gala
Oaxaca","Hacienda Los Laureles","Hacienda La Noria","Holiday Inn Express","Kabii","La
Casona de tita","La Casona del Llano","Las Golondrinas","Las Mariposas","Las Mercedes","Los
Olivos","Los Pilares","Marqués del Valle","Misión de los Angeles","Oaxaca Dorado","Oaxaca
Guest House","Oaxaca Inn Express","Oaxaca Magico","Oaxaca Real","One","Parador
Crespo","Parador De Alcala",
                    "Parador Del Domico","Parador San Agustin","Parador San Andres","Quinta
Real Oaxaca","Real De antequera","Real Del Fortin","Real Santo
Domingo","Rivera","Trebol","Victoria","Villa de Leon","Villa Oaxaca","Villa Vazari","Villas del
Sol","Xadani")

rownames(DatosHotelR) <- c("Turista Deportivo", "Turista Cultural", "Turista Negocios",
"Turista Sol y Playa", "Turista de Costumbres y Tradiciones","Turista Academico
Cientifico","Turista de Aventura Ecoturismo","Turista Convencional","Turista Social")
valoraciones <- as(DatosHotelR, "matrix")
gsub(".","",valoraciones)
```

```
valoraciones <- as.data.frame(valoraciones)
library(tidyverse)
valoraciones <- valoraciones %>% rownames_to_column(var = "usuario")
valoraciones_tidy <- valoraciones %>% gather(key = "Hoteles",
                                             value = "valoracion",
                                             -usuario)



total_NA <- valoraciones %>% select(-usuario) %>%
  map_dbl(.f = function(x){ sum(is.na(x))}) %>%
  reduce(.f = sum)
total_elementos <- (ncol(valoraciones) - 1) * (nrow(valoraciones))
porcentaje_NA   <- 100 * (total_NA / total_elementos)
porcentaje_NA

valoraciones_tidy %>% filter(!is.na(valoracion)) %>% select(valoracion) %>%
group_by(valoracion) %>% count() %>%
  ggplot(aes(x = valoracion, y = n)) +
  geom_col(color = "black") +
  theme_bw()


valoraciones_tidy <- valoraciones_tidy %>%
  group_by(usuario) %>%
  mutate(valoracion = scale(valoracion)) %>%
  ungroup()

usuarios_excluidos <- valoraciones_tidy %>% filter(!is.na(valoracion)) %>%
  group_by(usuario) %>% count() %>% filter(n < 2) %>%
  pull(usuario)
usuarios_excluidos

valoraciones_tidy <- valoraciones_tidy %>% filter(!usuario %in% usuarios_excluidos)
valoraciones_usuarios <- valoraciones_tidy %>%
  spread(key = usuario, value = valoracion, fill = NA)

funcion_correlacion <- function(x, y){
  correlacion <- cor(x, y, use = "na.or.complete", method = "pearson")
  return(correlacion)
}


similitud_usuarios <- map_dbl(.x = valoraciones_usuarios[, -1],
                              .f = funcion_correlacion,
                              y = valoraciones_usuarios[,2])
```

```
similitud_usuarios <- data_frame(usuario   = names(similitud_usuarios),
                                 similitud = similitud_usuarios) %>%
  arrange(desc(similitud))
head(similitud_usuarios)



hoteles_visitados<- valoraciones_tidy %>%
  filter(usuario == tipo & !is.na(valoracion)) %>%
  pull(Hoteles)
hoteles_visitados

hoteles_no_visitados <- valoraciones_tidy %>%
  filter(usuario == tipo & is.na(valoracion)) %>%
  pull(Hoteles)
hoteles_no_visitados

prediccion <- rep(NA, length(hoteles_no_visitados))
hotel   <- rep(NA, length(hoteles_no_visitados))
n_obs_prediccion <- rep(NA, length(hoteles_no_visitados))



for(i in seq_along(hoteles_no_visitados)){
  # Usuarios que han visitado el hotel i
  usuarios_hotel_i <- valoraciones_tidy %>%
    filter(Hoteles == hoteles_no_visitados[i] &
           !is.na(valoracion)) %>% pull(usuario)
  # Si no hay un mínimo de usuarios que han visitado el hotel, no se considera una
  # estimación suficientemente buena por lo que se pasa a la siguiente.
  if (length(usuarios_hotel_i) < 2){
    next()
  }
  # Los 10 usuarios más parecidos de entre los que han visitado el hotel i, cuya
  # similitud es >= 0.
  top_10_usuarios <- similitud_usuarios %>%
    filter(similitud >= 0 & (usuario %in% usuarios_hotel_i)) %>%
    arrange(desc(similitud)) %>%
    head(10)
  # Si no hay un mínimo de usuarios con valoraciones válidas, no se considera una
  # estimación suficientemente buena por lo que se pasa a la siguiente .
  if (nrow(top_10_usuarios) < 2){
    next()
  }
  # Valoraciones de esos 10 usuarios sobre la película i
  valoraciones_top_10 <- valoraciones_tidy %>%
```

```
  filter(Hoteles == hoteles_no_visitados[i] &
         usuario %in% top_10_usuarios$usuario)
 # Media ponderada de las valoraciones de los top_10_usuarios
 top_10_usuarios <- top_10_usuarios %>% left_join(valoraciones_top_10,
                                                  by = "usuario")
 prediccion[i] <- sum(top_10_usuarios$similitud * top_10_usuarios$valoracion) /
  sum(top_10_usuarios$similitud)
 hotel[i] <- hoteles_no_visitados[i]
 n_obs_prediccion[i] <- nrow(top_10_usuarios)
}
top5_recomendaciones <- data.frame(hotel, prediccion, n_obs_prediccion) %>%
 arrange(desc(prediccion)) %>%
 head(5)
top5_recomendaciones
write.csv(top5_recomendaciones, file="Prueba1.csv" , row.names = F)
hotelesvistadosframe <- data.frame(hoteles_visitados)
write.csv(hotelesvistadosframe, file="HotelesVisitados.csv" , row.names = F)
dev.off()
```

3.8.3 Módulo de Resultados de Recomendación de Productos Turísticos

(View)

```
<?php
$hotel1 = array(
        'name'  => 'hotel1',
        'id'  => 'hotel1',
        'value' => set_value('hotel1'),
        'maxlength' => 80,
        'size'  => 50,
);
$submitr = array(
        'name' => 'submit',
        'id' => 'submit',
        'value' => 'Enviar Busqueda',
        'title' => 'Enviar Busqueda'
);



?>
<?php  $this->load->view('inicioUF/cabecera');?>
<?php  $this->load->view('inicioUF/body');?>

<!-- end header -->
<section id="inner-headline">
        <div class="container">
                <div class="row">
                        <div class="col-lg-12">
```

```
                                <ul class="breadcrumb">
                                        <li><a
href="<?=base_url()?>index.php/Welcome/index"><i class="fa fa-home"></i></a><i
class="icon-angle-right"></i></li>
                                        <li class="active">Resultados de
Busqueda</li>
                                </ul>
                        </div>
                </div>
        </div>
</section>


<section id="content">
        <div class="container">
                <div class="row">
                        <div class="col-lg-6">
                                <div class="searchtitle">
                                        <p class="lead">Sugerencias Segun Perfil
Turistico</p>
                                </div>
                                <article>
                                        <div class="post-image">
                                                <div class="post-heading">
                                                        <h3><a href="#">Hoteles
Sugeridos</a></h3>
                                                </div>

                                                <div id="main-slider"
class="main-slider flexslider">
                                                        <ul class="slides">
                                                                <li>
                                                                        <img
style="width:550px; height:400px;" id="myImg1"/>
                                                                        <div
class="flex-caption">

<h2 id="nombre1"></h2>

<label name="ubicacion1" id="ubicacion1"> </label>

<a onclick="ubicar1()" class="btn btn-theme">Ver en el mapa</a>
                                                                        </div>
                                                                </li>
                                                                <li>
```

```
                                                                    <img
style="width:550px; height:400px;" id="myImg2"/>
                                                                    <div
class="flex-caption">

<h2 id="nombre2"></h2>

<p name="ubicacion2" id="ubicacion2"></p>

<a onclick="ubicar2()" class="btn btn-theme">Ver en el mapa</a>
                                                                    </div>
                                                            </li>

                                                            <li>
                                                                    <img
style="width:550px; height:400px;" id="myImg3"/>
                                                                    <div
class="flex-caption">

<h2 id="nombre3"></h2>

<label name="ubicacion3" id="ubicacion3"></label>

<a onclick="ubicar3()" class="btn btn-theme">Ver en el mapa</a>
                                                                    </div>
                                                            </li>

                                                            <li>
                                                                    <img
style="width:550px; height:400px;" id="myImg4"/>
                                                                    <div
class="flex-caption">

<h2 id="nombre4"></h2>

<label name="ubicacion4" id="ubicacion4"></label>

<a onclick="ubicar4()" class="btn btn-theme">Ver en el mapa</a>
                                                                    </div>
                                                            </li>

                                                            <li>
                                                                    <img
style="width:550px; height:400px;" id="myImg5"/>
```

```
                                                                        <div
class="flex-caption">

<h2 id="nombre5"></h2>

<label name="ubicacion5" id="ubicacion5"></label>

<a onclick="ubicar5()" class="btn btn-theme">Ver en el mapa</a>
                                                                        </div>
                                                                </li>
                                                        </ul>
                                                </div>
                                        </div>
                                </article>
                                <article>
                                        <div class="post-image">
                                                <div class="post-heading">
                                                        <h3><a
href="#">Restaurantes</a></h3>
                                                </div>

                                                <div id="main-slider"
class="main-slider flexslider">
                                                        <ul class="slides">
                                                                <li>
                                                                        <img
style="width:550px; height:400px;" id="rmyImg1"/>
                                                                        <div
class="flex-caption">

<h2 id="nombrer1"></h2>

<h3 name="ubicacion1r" id="ubicacion1r"></h3>

<a href="#" class="btn btn-theme">Learn More</a>
                                                                        </div>
                                                                </li>
                                                                <li>
                                                                        <img
style="width:550px; height:400px;" id="rmyImg2"/>
                                                                        <div
class="flex-caption">

<h2 id="nombrer2"></h2>
```

```
<h3 name="ubicacion2r" id="ubicacion2r"></h3>

<a href="#" class="btn btn-theme">Learn More</a>
                                                                        </div>
                                                                </li>

                                                                <li>
                                                                        <img
style="width:550px; height:400px;" id="rmyImg3"/>
                                                                        <div
class="flex-caption">

<h2 id="nombrer3"></h2>

<h3 name="ubicacion3r" id="ubicacion3r"></h3>

<a href="#" class="btn btn-theme">Learn More</a>
                                                                        </div>
                                                                </li>

                                                                <li>
                                                                        <img
style="width:550px; height:400px;"  id="rmyImg4"/>
                                                                        <div
class="flex-caption">

<h2 id="nombrer4"></h2>

<h3 name="ubicacion4r" id="ubicacion4r"></h3>

<a href="#" class="btn btn-theme">Learn More</a>
                                                                        </div>
                                                                </li>

                                                                <li>
                                                                        <img
style="width:550px; height:400px;" id="rmyImg5"/>
                                                                        <div
class="flex-caption">

<h2 id="nombrer5"></h2>

<h3 name="ubicacion5r" id="ubicacion5r"></h3>
```

```
<a href="#" class="btn btn-theme">Learn More</a>
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                
</div>
</li>
</ul>
</div>
</div>

</article>


<article>
<h3>Sitios en Google Maps</h3>
<script>
var map;
var geocoder;
function initMap() {
map = new
google.maps.Map(document.getElementById('map'), {
center: {lat:
17.0719328, lng: -96.7547896},
zoom: 17
});
geocoder = new
google.maps.Geocoder();
for (var i = 0; i <
marcadores.length; i++) {


geocodeAddress(geocoder, map, marcadores[i]);

}

}



function geocodeAddress(geocoder,
resultsMap, ubicacion) {

var address = ubicacion ;

geocoder.geocode({'address': address}, function(results, status) {
if (status ===
'OK') {
```

```
resultsMap.setCenter(results[0].geometry.location);
                                                                    var
infowindow = new google.maps.InfoWindow({

content: ubicacion
                                                                    });

                                                                    var
marker = new google.maps.Marker({

map: resultsMap,

position: results[0].geometry.location
                                                                    });


marker.addListener('click', function() {

infowindow.open(map, marker);
                                                                    });



resultsMap.fitBounds(bounds);

resultsMap.panToBounds(bounds);
                                                            } else {

alert('Geocode was not successful for the following reason: ' + status);
                                                            }
                                                    });
                                            }


                                            function ubicar1(){
                                                    initMap();
                                                    geocodeAddress(geocoder,
map, document.getElementById('ubicacion1').innerText);
                                            }
                                            function ubicar2(){
                                                    initMap();
                                                    geocodeAddress(geocoder,
map, document.getElementById('nombre2').innerText);
                                            }
```

```
                                                function ubicar3(){
                                                        initMap();
                                                        geocodeAddress(geocoder,
map, document.getElementById('ubicacion3').innerText);
                                                }
                                                function ubicar4(){
                                                        initMap();
                                                        geocodeAddress(geocoder,
map, document.getElementById('ubicacion4').innerText);
                                                }
                                                function ubicar5(){
                                                        initMap();
                                                        geocodeAddress(geocoder,
map, document.getElementById('ubicacion5').innerText);
                                                }
                                                function ubicarenmapa(nombre){
                                                        initMap();
                                                        geocodeAddress(geocoder,
map, nombre);

                                                }

                                        </script>


                                </article>
                        </div>

                        <div class="col-lg-4">
                                <aside class="right-sidebar">
                                        <div class="widget">
                                                <form role="form">
                                                        <div class="form-group">
                                                                <input
type="text" class="form-control" id="s" placeholder="Search..">
                                                        </div>
                                                </form>
                                        </div>
                                        <div class="widget">
                                                <h5 class="widgetheading">Google
Maps</h5>


                                                <div id="map">
```

```
                                                        </div>
                                                </div>


                                                <div class="widget">



                                                        <h5 class="widgetheading">Hoteles
Recomendados</h5>

                                                        <ul class="tags">

                                                                <?php
                                                                if ($hoteles_visitados !=
FALSE){
                                                                        foreach
($hoteles_visitados as $hotel=>$nombre){
                                                                                ?>
                                                                                <li><a
onclick="ubicarenmapa('<?php echo $nombre;?>')"><?php echo $nombre?></a></li>
                                                                                <?php
                                                                        }
                                                                }

                                                                ?>
                                                        </ul>


                                                        <div>

                                                        </div>

                                                </div>
                                                <div class="widget">
                                                        <h5
class="widgetheading">Restaurantes Recomendados</h5>
                                                        <ul class="tags">
                                                                <?php
                                                                if
($restaurantes_visitados != FALSE){
                                                                        foreach
($restaurantes_visitados as $restaurante=>$nombre){
                                                                                ?>
```

```
<li></i><a onclick="ubicarenmapa('<?php echo $nombre;?>')"><?php echo
$nombre?></a></li>
                                                                                <?php
                                                                        }
                                                                }

                                                                ?>

                                                        </ul>
                                                </div>

                                        </aside>
                                </div>

                        </div>


                </div>
        </section>

        <footer>
                <div class="container">
                        <div class="row">

                                <div class="col-lg-3">
                                        <div class="widget">
                                                <img style="width:250px; height:200px;"
src="<?php echo base_url()?>assets/images/icono_appsituroax.png" />
                                        </div>
                                </div>

                                <div class="col-lg-3">
                                        <div class="widget">
                                                <h4>SiturOax</h4>
                                                <address>
                                                        Tecnológico Nacional de México /
Instituto Tecnológico de Oaxaca Avenida Ing. Víctor Bravo Ahuja No. 125 Esquina
Calzada Tecnológico, C.P. 68030
                                                </address>

                                        </div>
                                </div>
                                <div class="col-lg-3">
                                        <div class="widget">
```

```
                                        <h4>Informacion</h4>
                                        <address>
                                                <p>Sitio realizado el Tecnológico
Nacional de México / Instituto Tecnológico de Oaxaca.</p>
                                                <p> Departamento de Posgrado e
Investigacion </p>

                                        </address>
                                </div>
                        </div>

                </div>
        </div>

</footer>

</div>

<script type="text/javascript">

        var jsonhotel = <?php echo $json;?> ;
        var jsonrestaurante = <?php echo $json2;?> ;

        for (var i = 0; i < jsonhotel.length-1; i++) {
                console.log("Datos del estudiante "+(i+1));
                console.log("Hotel: "+jsonhotel[i].hotel);
                //document.getElementById("hotel" + (i+1)).value =
jsonhotel[i].hotel;
                //document.getElementById("restaurante" + (i+1)).value =
jsonrestaurante[i].restaurante;
                document.getElementById("nombrer" + (i+1)).textContent =
jsonrestaurante[i].restaurante;
                document.getElementById("nombre" + (i+1)).textContent =
jsonhotel[i].hotel;
                document.getElementById("myImg"+ (i+1)).src = "<?php echo
base_url()?>assets/images/HotelesOaxaca/"+ jsonhotel[i].hotel+".jpg";
                document.getElementById("rmyImg"+ (i+1)).src = "<?php echo
base_url()?>assets/images/RestaurantesOaxaca/"+
jsonrestaurante[i].restaurante+".jpg";




        }
```

```
        var marcadores = ['<?php echo $ubicacion1;?>', '<?php echo
$ubicacion2;?>', '<?php echo $ubicacion3;?>', '<?php echo $ubicacion4;?>',
'<?php echo $ubicacion5;?>'];

        document.getElementById("ubicacion1").textContent = '<?php echo
$ubicacion1;?>';
        document.getElementById("ubicacion2").textContent = '<?php echo
$ubicacion2;?>';
        document.getElementById("ubicacion3").textContent = '<?php echo
$ubicacion3;?>';
        document.getElementById("ubicacion4").textContent = '<?php echo
$ubicacion4;?>';
        document.getElementById("ubicacion5").textContent = '<?php echo
$ubicacion5;?>';

</script>

<a href="#" class="scrollup"><i class="fa fa-angle-up active"></i></a>
<?php  $this->load->view('inicioUF/final');?>\
```

3.8.4 Módulo de Evaluación de Productos Turísticos

(View)

```
    <!-- end header -->
    <section id="inner-headline">
     <div class="container">
      <div class="row">
       <div class="col-lg-12">
        <ul class="breadcrumb">
         <li><a href="<?=base_url()?>index.php/Welcome/index"><i class="fa
fa-home"></i></a><i class="icon-angle-right"></i></li>
         <li class="active">Evaluacion</li>
       </ul>
     </div>
   </div>
 </div>
</section>
<section id="content">
 <div class="container">
  <form action= "<?php echo base_url()?>index.php/turista/calificar"
method="post" accept-charset="utf-8">
```

```
    <div class="row">
     <h2>Evaluar Hotel <small> visitado</small></h2>

     <div class="col-xs-12 col-sm-2 col-md-8 col-sm-offset-4 col-md-offset-2">
      <form role="form" class="register-form">
       <h2>Lista de Hoteles Sugeridos</h2>

       <hr class="colorgraph">
       <div class="form-group">

        <!-- tile body -->
        <div class="tile-body nopadding">

         <div class="table-responsive">
           <table  class="table table-datatable table-bordered"
id="inlineEditDataTable">
             <thead>
               <tr>
                 <th class="sort-alpha">ID Recomendacion</th>
                 <th class="sort-alpha">Fecha de la Busqueda Realizada</th>
               </tr>
             </thead>
             <tbody>
               <?php
               if ($recomendaciones != FALSE){
                 foreach ($recomendaciones as $recomendacion){
                   echo "<tr>";
                  echo "<td><a name='idrec' id='idrec' href='".base_url().
"index.php/turista/evaluarrecomendacion/".$recomendacion->id_recomendacion."'>
$recomendacion->id_recomendacion</a></td>";
                   echo "<td>".$recomendacion->fecha_busqueda."</td>";
                   echo "</tr>";
                 }
               }
               ?>
             </tr>
           </tbody>
         </table>
       </div>

      </div>


      <?php
      foreach ($una_recomendacion as $recomendacion) {
```

```
        ?>
        <input style="display: none" type="text" name="idr" id="idr"
value="<?php echo $recomendacion->id_recomendacion; ?>">
        <select name="hoteleseval" id="hoteleseval" class="form-control
input-lg">
          <option><?php echo $recomendacion->hotel1; ?></option>
          <option><?php echo $recomendacion->hotel2; ?></option>
          <option><?php echo $recomendacion->hotel3; ?></option>
          <option><?php echo $recomendacion->hotel4; ?></option>
          <option><?php echo $recomendacion->hotel5; ?></option>
        </select>
        <?php
      }

      ?>



      <script >

        $('#hoteleseval').change(function(){
          var objDiv = document.getElementById("dis");
          objDiv.innerHTML="<h3>Recepcion <small> del Hotel</small></h3><br>";
          objDiv.innerHTML=objDiv.innerHTML+"<label><input type='radio'
class='radio-inline' name='radios' value='1'><span class='outside'><span
class='inside'></span></span>1</label>";
          objDiv.innerHTML=objDiv.innerHTML+"<label><input type='radio'
class='radio-inline' name='radios' value='2'><span class='outside'><span
class='inside'></span></span>2</label>";
          objDiv.innerHTML=objDiv.innerHTML+"<label><input type='radio'
class='radio-inline' name='radios' value='3'><span class='outside'><span
class='inside'></span></span>3</label>";
          objDiv.innerHTML=objDiv.innerHTML+"<label><input type='radio'
class='radio-inline' name='radios' value='4'><span class='outside'><span
class='inside'></span></span>4</label>";
          objDiv.innerHTML=objDiv.innerHTML+"<label><input type='radio'
class='radio-inline' name='radios' value='5'><span class='outside'><span
class='inside'></span></span>5</label>";
          objDiv.innerHTML=objDiv.innerHTML+"<label><input type='radio'
class='radio-inline' name='radios' value='6'><span class='outside'><span
class='inside'></span></span>6</label>";
          objDiv.innerHTML=objDiv.innerHTML+"<label><input type='radio'
class='radio-inline' name='radios' value='7'><span class='outside'><span
class='inside'></span></span>7</label>";
```

(Controllers)

```
public function realizarevaluacion(){

		$filename =
'C:/xampp/htdocs/Residencia2018/assets/ejecutaR/HotelesVisitados.csv';
		$csvArray = $this->ImportCSV2Array($filename);

		foreach ($csvArray as $row)
		{
			$resultado[] = $row['hoteles_visitados'];
		}
		$data['hoteles_visitados'] = $resultado;


		$this->load->view('evaluaciones/evaluacionproductos', $data);

	}

	public function evaluarrecomendacion(){
		$this->load->model('turistaModel');
		$id_user = $this->tank_auth->get_user_id();
		$id = $this->uri->segment(3);
		$data['recomendaciones'] = $this->turistaModel->verRecomendacion($id_user);
		$data['una_recomendacion'] = $this->turistaModel->verunaRecomendacion($id);
		$this->load->view('evaluaciones/evaluacionrecomendacion',$data);

	}


	public function evaluarRestaurante(){
		$this->load->model('turistaModel');
		$id_user = $this->tank_auth->get_user_id();
		$id = $this->uri->segment(3);
		$data['recomendaciones'] = $this->turistaModel->verRecomendacion($id_user);
		$data['una_recomendacion'] = $this->turistaModel->verunaRecomendacion($id);
		$this->load->view('evaluaciones/evaluacionrestaurante',$data);

	}


	public function calificar(){
		$this->load->model('turistaModel');
```

```
$id_user = $this->tank_auth->get_user_id();
$id = $this->uri->segment(3); //id de recomendacion

$nombre_hotel = $this->input->post('hoteleseval');
$cal1 = $this->input->post('radios');
$cal2 = $this->input->post('radios2');
$cal3 = $this->input->post('radios3');
$cal4 = $this->input->post('radios4');
$cal5 = $this->input->post('radios5');
$cal6 = $this->input->post('radios6');
$cal7 = $this->input->post('radios7');
$rec = $this->input->post('idr');

$cal_final = (($cal1 + $cal2 + $cal3 + $cal4 + $cal5 + $cal6 + $cal7)/7);


$result = $this->db->query('select tipo_perfil FROM busqueda INNER JOIN
recomendacion ON busqueda.id_busqueda = recomendacion.id_busqueda WHERE
recomendacion.id_recomendacion='.$rec);

$result2 = $result->row_array();
$pp = $result2['tipo_perfil'];

$data['recomendaciones'] = $this->turistaModel->verRecomendacion($id_user);
$data['una_recomendacion'] = $this->turistaModel->verunaRecomendacion($id);


$query_cal = "insert into calificaciones_hotel values
('$rec','$nombre_hotel','$cal_final','$pp')";
$this->db->query($query_cal);

exec("Rscript C:/xampp/htdocs/Residencia2018/assets/ejecutaR/calificarHotel.R
\"$pp\" \"$nombre_hotel\"  \"$cal_final\" ");

$this->load->view('evaluaciones/evaluacionrecomendacion',$data);


}
```

4.9 Fase 4 Pruebas

Al realizar las pruebas de búsquedas de servicios turísticos se obtuvieron como resultados las recomendaciones de hoteles y restaurantes.

Formulario el cual requiere de los parámetros de búsqueda. Seleccionando las preferencias.

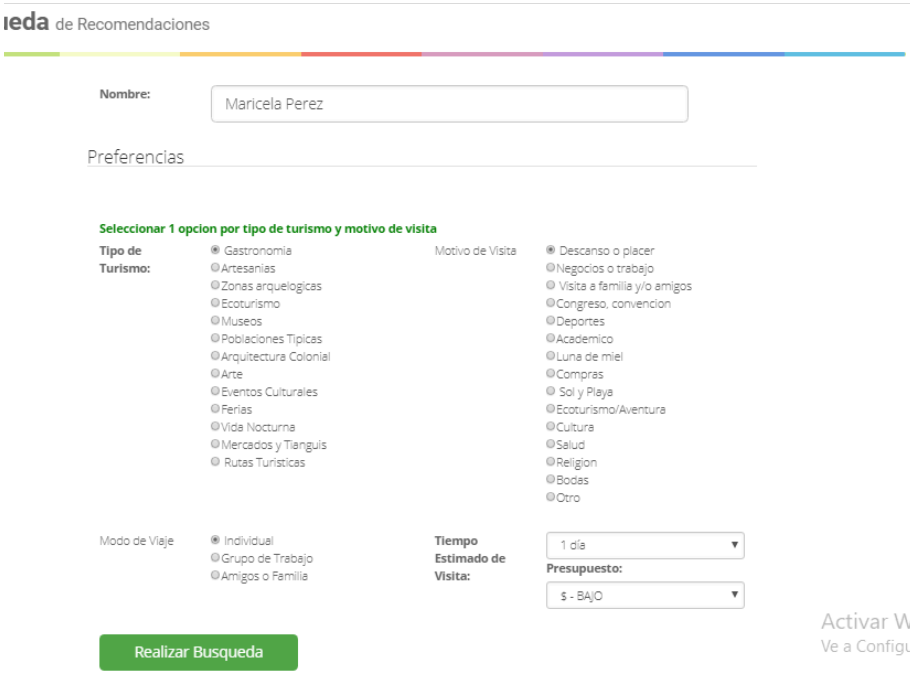


Ilustración 25 Prueba de Búsqueda 1

Resultados de la búsqueda

El algoritmo se ejecuta con los parámetros seleccionados en el formulario de búsqueda. Considera el motivo de visita y el tipo de turismo del usuario. En este caso al seleccionar el motivo de visita “Descanso o placer”, el controlador lo toma para hacer la clasificación del perfil, como ya vimos en la codificación la clasificación entra en el “perfil convencional”.

La matriz de calificaciones que maneja R para hacer la sugerencia considera los demás perfiles para hacer sugerencias a este otro perfil. Los resultados son los siguientes.

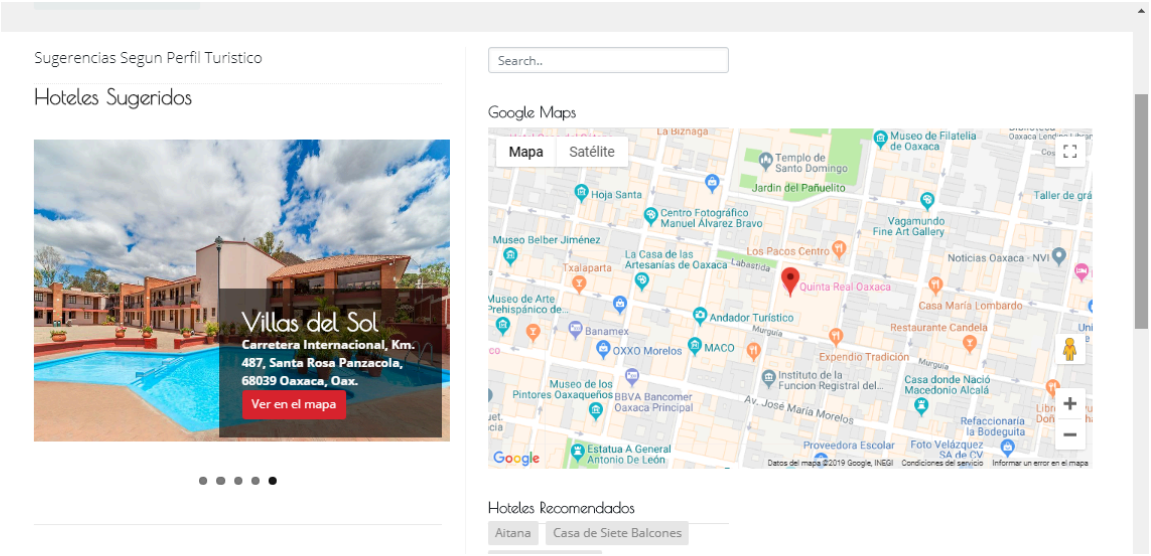


Ilustración 26Prueba de búsqueda 2

La ejecución correcta del algoritmo genera archivos y las cuales podemos verificar

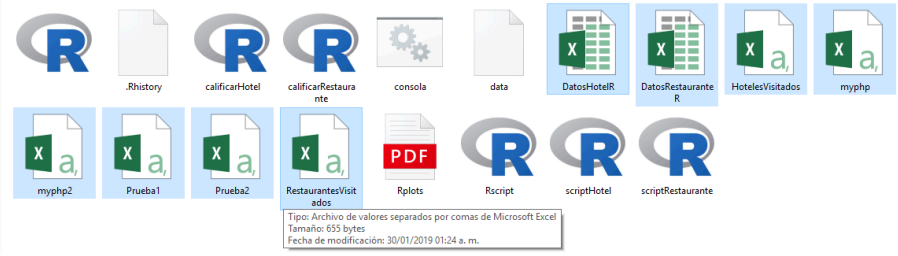


Ilustración 27 Prueba de búsqueda 3

Además, corroboramos la inserción correcta de la información sobre la búsqueda y recomendación a la base de datos para retroalimentar la matriz y generar históricos que sirven para la toma de decisiones del sistema como tal.

4.9 Fase 1: Planeación

Para el levantamiento de requerimientos de la tercera iteración se tuvo una reunión con el cliente en la que se trataron asuntos relacionados con la presentación de la información para la toma de decisiones (dashboard).

3.9.1Historias de usuario

Se generaron las siguientes historias de usuario

Historia de usuario	
Numero: 9	Usuario: Administrador
Nombre de la historia: Dashboard	
Prioridad: alta	Riesgo de desarrollo: alto
Puntos estimados: 5	Iteración Asignada:3
Programador Responsable: Maricela	
Descripción: se elaborara la interfaz del dashborad para la visualización de la información correspondiente sobre los productos turísticos y así los usuarios interesados puedan realizar de una manera más fácil la toma de decisiones	
Observaciones:	

4.10 Fase 2: Diseño

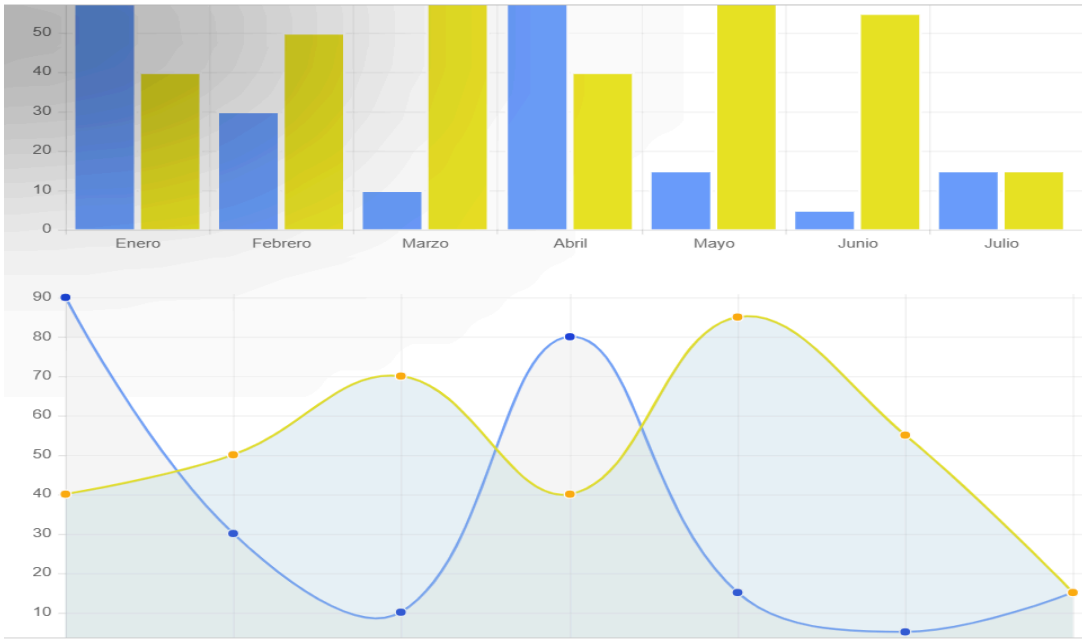


Ilustración 28 Interfaz de dashboard

4.11 Codificación

```
function dashboard(){
        $data['user_id']        = $this->tank_auth->get_user_id();
        $data['usernombre']     = $this->tank_auth->get_username();


        $result = $this->db->query('select id, username, email from users where id =
'.$this->tank_auth->get_user_id());
        $result = $result->result_array();
        $data['usuario'] = $result[0];

        $con1 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Descanso o placer"');
        $row1 = $con1->row_array();
        $data['descanso'] =  $row1['total'] ;

        $con2 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Negocios o trabajo"');
        $row2 = $con2->row_array();
        $data['negocios'] =  $row2['total'] ;

        $con3 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Visita a familia y/o amigos"');
        $row3 = $con3->row_array();
```

```
$data['familia'] =  $row3['total'] ;

$con4 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Congreso-convencion"');
$row4 = $con4->row_array();
$data['congreso'] =  $row4['total'] ;


$con5 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Deportes"');
$row5 = $con5->row_array();
$data['deportes'] =  $row5['total'] ;

$con6 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Academico"');
$row6 = $con6->row_array();
$data['academico'] =  $row6['total'] ;

$con7 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Luna de miel"');
$row7 = $con7->row_array();
$data['luna'] =  $row7['total'] ;

$con8 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Compras"');
$row8 = $con8->row_array();
$data['compras'] =  $row8['total'] ;

$con9 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Sol y Playa"');
$row9 = $con9->row_array();
$data['sol'] =  $row9['total'] ;

$con10 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Ecoturismo/Aventura"');
$row10 = $con10->row_array();
$data['ecoturismo'] =  $row10['total'] ;


$con11 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Cultura"');
$row11 = $con11->row_array();
$data['cultura'] =  $row11['total'] ;

$con12 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Salud"');
```

```
$row12 = $con12->row_array();
$data['salud'] =  $row12['total'] ;


$con13 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Religion"');
$row13 = $con13->row_array();
$data['religion'] =  $row13['total'] ;


$con14 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Bodas"');
$row14 = $con14->row_array();
$data['bodas'] =  $row14['total'] ;

$con15 = $this->db->query('select  motivovisita, COUNT(*) as total FROM
busqueda WHERE busqueda.motivovisita = "Otro"');
$row15 = $con15->row_array();
$data['otro'] =  $row15['total'] ;

$fechas = $this->db->query('select COUNT(*) as total ,
YEAR(`fecha_busqueda`) as anio FROM `recomendacion` GROUP BY
YEAR(`fecha_busqueda`)');
$fechas = $fechas->result();
$data['fechasanio'] = $fechas;



$genero = $this->db->query('select count(id) as total_usuarios,
count(IF(genero="Masculino",genero,NULL)) as Masculino,
count(IF(genero="Femenino",genero,NULL)) as Femenino from users_final order by id');
$genero = $genero->row_array();
$data['genero'] = $genero;


$this->load->view('dashboard/verdash', $data);
}
```

4.11 Pruebas

Como se puede observar en la imagen siguiente del lado izquierdo se muestran por año el número de búsquedas realizadas. Del lado derecho en la gráfica de pastel se muestran el género que realizo más búsquedas.

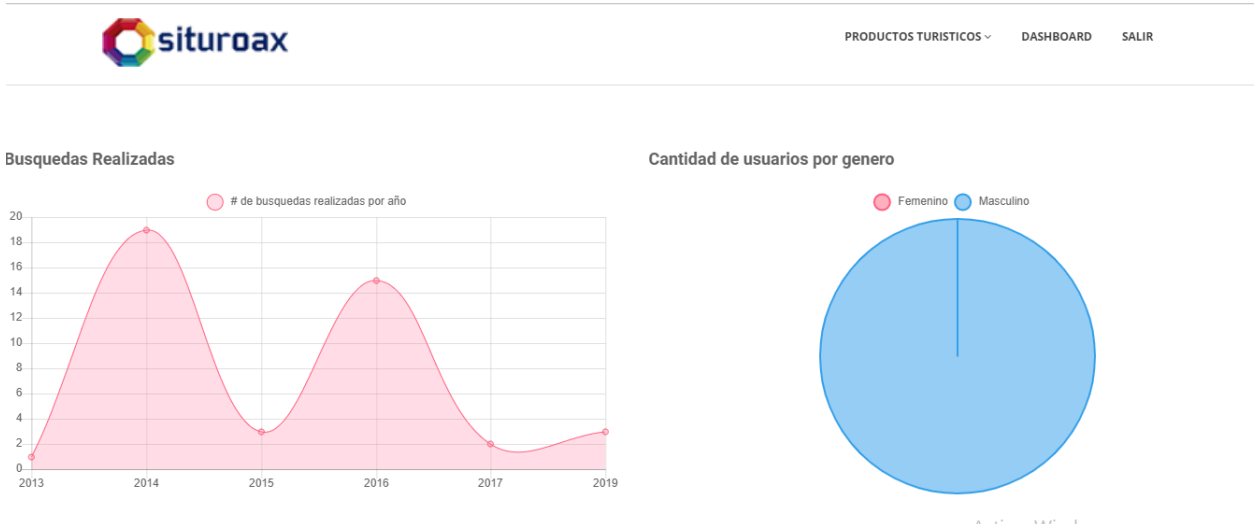


Ilustración 29 Dashboard

En la siguiente grafica se muestran los motivos de visita que más se han presentado al realizar las búsquedas.

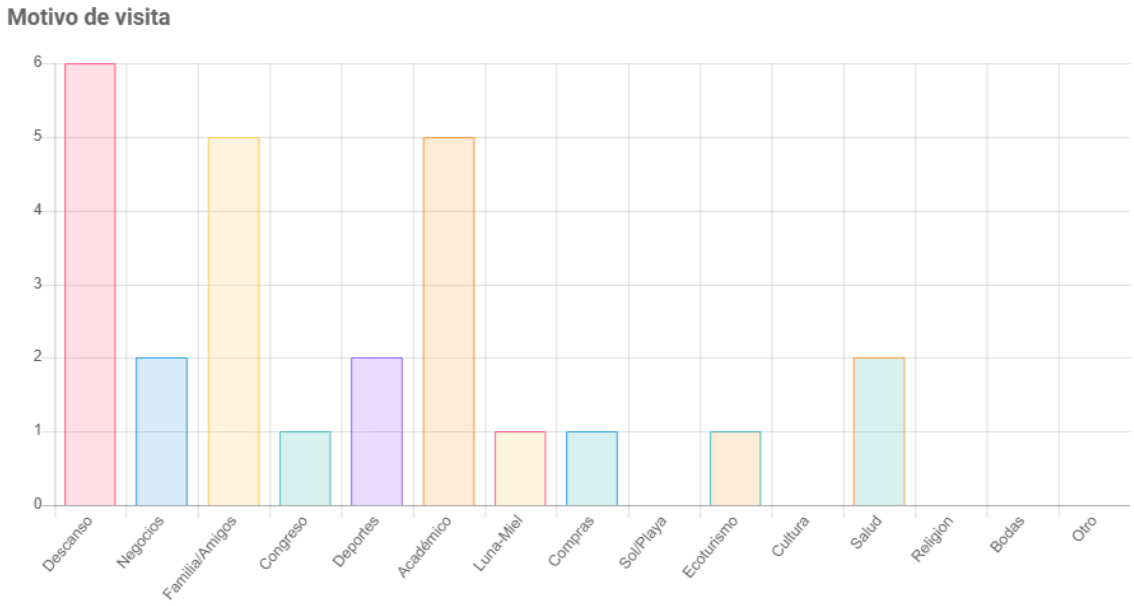


Ilustración 30 Tabla en barra del Dashboard

4.12 Pruebas de hipótesis

Con el objeto de comprobar la hipótesis establecida en el presente documento se implementó el algoritmo de filtrado colaborativo para la búsqueda de servicios turísticos, donde se establecieron al sistema diversas pruebas de los diferentes perfiles turísticos.

Para la aplicación de este algoritmo fue requerido realizar consultas a la base de datos de encuestas, para obtener una matriz de promedio de las calificaciones realizadas a partir de las percepciones de los turistas.

Dicho promedio va de 0 a 7 donde el valor de 0 es una mala percepción del servicio calificado por el turista y 7 es excelente calificación.

Misión.de.	Oaxaca.Dc	Oaxaca.Gu	Oaxaca.Inr	Oaxaca.Ma	Oaxaca.Re	One	Parador.Cr	Parador.De	Parador.De	Parador.Sa	Parador.Sa	C
								7				
6.6333			6.7333	6.6	5.1333	6.3333	6.6	5.9333	7		7	
7	6.8333		6.2	5.9333	5.1333	5.6	4.9667		7	6.8333	7	
								5.2667				
				5.5333								
6.2333					6.1333						5	
7												
7	6.7667	5.9333	3.8333						6.1	7		
5.6			5.8333		5.6	6	6.0667	6.8	7		4.8667	

Ilustración 31 Tebla de promedios

Después de obtener la matriz se predice la valoración que un determinado perfil hará sobre un producto turístico utilizando las valoraciones que han hecho sobre ese mismo producto los *n* perfiles turísticos más parecidos a él. La similitud entre perfiles se mide acorde al patrón de valoraciones que tiene cada uno, en este caso, las filas de la matriz.

En primer lugar, se calcula la similitud entre un perfil turístico y el resto de perfiles. En este caso, se emplea como medida de similitud el coeficiente de correlación de Pearson. Como los perfiles están definidos por las filas, hay que transponer la matriz para realizar los cálculos.

promedios × | matrizCalRestaurantes × | DatosHotelR ×

Filter

	usuario	similitud
1	Turista Academico Cientifico	1.00000000
2	Turista Cultural	0.51515468
3	Turista Social	0.45774652
4	Turista de Costumbres y Tradiciones	0.35518770
5	Turista de Aventura Ecoturismo	0.31440916
6	Turista Convencional	0.18477393
7	Turista Negocios	-0.06518269
8	Turista Deportivo	-0.55779703
9	Turista Sol y Playa	-1.00000000

Ilustración 32 Similitud de perfiles

Una vez ordenados los perfiles de mayor a menor similitud respecto al perfil x se procede a calcular la predicción de la valoración.

	Hoteles	Turista Academico Cientifico	Turista Convencional	Turista Cultural	Turista de Aventura Ecoturismo	Turista de Costumbres y Tradiciones	Turista Deportivo	Turista Negocios	Turista Social	Turista Sol y Playa
1	Abu	NA	0.12493885	1.91458696	NA	-1.0692348	NA	-4.43875374	NA	NA
2	Aitana	-1.11078575	0.74068428	0.63498723	0.72816692	NA	1.2287481	0.37502878	1.11860084	0.2091268
3	Angel Inn	NA	-0.11448350	0.27511106	NA	NA	NA	-0.66893034	-0.76876901	NA
4	Anua	NA	NA	NA	0.58882864	NA	NA	NA	NA	NA
5	Aurora	0.96728353	-0.45653008	0.75494596	NA	NA	NA	NA	NA	NA
6	Azucenas	NA	-0.90130353	0.83495843	NA	NA	NA	0.75207202	0.69396386	NA
7	Azul Cielo	1.04428754	0.91175889	-0.52453381	NA	NA	NA	NA	NA	NA
8	Azul Oaxaca	NA	0.53543581	NA	NA	NA	NA	NA	NA	NA

Ilustración 33 Correlación

Dado que se emplea la media ponderada como estimación final, no se pueden incluir pesos negativos. Como la correlación de Pearson toma valores en el rango [-1, +1], se emplean únicamente aquellas observaciones con valores mayores o iguales a cero. A efectos prácticos, equivale a decir que no se tienen en cuenta las valoraciones de los usuarios que tienen un perfil opuesto.

Aunque se establezca que se tienen que emplear los n perfiles más similares para predecir la valoración, puede ocurrir que, para algunos hoteles o restaurantes, no haya suficientes usuarios que las hayan valorado. Es conveniente recomendar únicamente hoteles y restaurantes cuya predicción esté basada en un mínimo de perfiles, de lo contrario la estimación puede ser muy mala.

Se consideran únicamente aquellos perfiles que han valorado al menos 15 hoteles y restaurantes.

Como resultado final se muestran como sugerencias los 5 hoteles y restaurantes con mayor valor de predicción.

	hotel	prediccion	n_obs_prediccion
1	Casa del Sotano Casa las Mercedes	1.0164641	2
2	Casantica	0.8552330	2
3	El Principal	0.7565966	2
4	Aurora	0.7481741	3
5	Las Mercedes	0.7253917	2

Ilustración 34 Recomenaciones

CAPÍTULO V. RESULTADOS Y CONCLUSIONES

Resultados de la investigación

Finalmente, en el desarrollo de este proyecto de tesis se analizaron y compararon algoritmos de recomendación, así como herramientas para desarrollar el sistema. La revisión y el análisis permitió conocer y estudiar el algoritmo de recomendación por filtrado colaborativo para realizar el modelado correcto y la implementación de este.

Los resultados obtenidos de dicha implementación fue la plataforma para la búsqueda y obtención de recomendaciones de servicios turísticos de los valles centrales del estado de Oaxaca. En esta plataforma el turista puede registrarse para tener acceso al sistema y poder ingresar sus datos para realizar la búsqueda y así darle una recomendación el sistema de los lugares, estos se pueden observar en un mapa de Google Maps para la ubicación y ayudar a la toma de decisión a los turistas.

Conclusión

El sistema desarrollado puede impactar en la manera en que los turistas toman decisiones basándose en el tipo de perfil que tiene, así como de sus intereses, presupuesto etc. muestra una interfaz amigable al usuario final que en este caso es el turista permitiéndole poder ingresar los datos que se necesitan para el buen funcionamiento.

Se halló que con el uso de este sistema se puede:

- Permitir la administración de nuevos productos turísticos su registro al sistema.
- Después de cambios realizados en los productos turísticos se actualizarán inmediatamente.
- Podrán darse de baja los productos turísticos.
- Ingresar datos para la recomendación de productos turísticos (hoteles, restaurantes).
- Mostar la información de los productos recomendados, en forma de lista, mostrando un mapa con su ubicación, imágenes un top 5 de recomendaciones.
- Podrán ser evaluados los hoteles y restaurante con el fin de que se puede retroalimentar el sistema.
- Se mostrará información relevante sobre algún producto o comparativos sobre productos turísticos mediante tablas.

Como producto comerciable “La Plataforma para la Captación, Difusión y Estudio de las preferencias, expectativas y precepciones de los turistas en Oaxaca” es una idea innovadora en el estado de Oaxaca ya que no se cuenta con un sistema así para los turistas.

Se tiene en cuenta que esta idea podría ser muy bien aceptada en otros estados del país.

Referencias

Castellano Torres, E. J. (Septiembre de 2007). EVALUACIÓN DEL USO DE ALGORITMOS COLABORATIVOS PARA ORIENTAR ACADÉMICAMENTE AL ALUMNADO EN BACHILLERATO. *Memoria Investigadora del segundo año de Doctorado*. Departamento de Informática, Universidad de Jaén.

Amat Rodrigo, J. (Marzo de 2018). *Sistemas de recomendación con R*. Obtenido de https://rpubs.com/Joaquin_AR/370301

Apache Friends. (s.f. de s.f. de 2002). *apachefriends.org*. Obtenido de https://www.apachefriends.org/es/download.html

Arteaga Camacho, J. G. (Junio de 2014). *Estudio comparativo de metodologías de desarrollo de software [Tabla]*. Recuperado el 04 de Febrero de 2019, de http://biblioteca.udenar.edu.co:8085/atenea/biblioteca/90255.pdf

Cobo, Á., Gómez, P., Pérez, D., & Rocha, R. (2015). *PHP Y MYSQL, Tecnologías para el desarrollo de aplicaciones web.* México: DIAZ DE SANTOS.

Contreras García , J. M., Molina Portillo, E., & Arteaga Cezón, P. (s.f.). *INTRODUCCION A LA PROGRAMACIÓN ESTADÍSTICA CON R PARA PROFESORES.* Mexico .

EllisLab. (2002-2018). *CodeIgniter ExpressionEngine ®*. Obtenido de https://codeigniter.com

Facultat d'Informàtica de Barcelona. (s.f. de s.f. de s.f.). *Intel.ligència Artificial*. Recuperado el 28 de Enero de 2019, de lsi.upc.es: http://www.lsi.upc.es/~bejar/ia/ia.html

Gilfillan, I. (2003). *La biblia de MySQL.* Anaya Multimedia.

Gómez Fuentes, M. (2013). *NOTAS DEL CURSO: BASES DE DATOS.* México: Publidisa Mexicana S. A. de C.V. Recuperado el Enero de 2019, de http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_del_curso_Bases_de_Datos.pdf

Google Maps Platform. (18 de Diciembre de 2018). *API Google Maps*. Recuperado el 04 de Febrero de 2019, de developers.google.com: https://developers.google.com/maps/documentation/javascript/tutorial

Hurwitz, J., & Kirsch, D. (2018). *Machine Learning For Dummies®, IBM Limited Edition.* River St, Hoboken: John Wiley & Sons, Inc.

Joskowicz, J. (10 de Febrero de 2008). Reglas y Prácticas en eXtreme Programming. Recuperado el 16 de Enero de 2019, de https://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf

Konyukhov, I. (s.f. de s.f. de 2018). *Tank Auth*. Obtenido de https://konyukhov.com/soft/tank_auth/

Leteliel, P., & Penadés, M. (s.f. de s.f. de s.f.). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* Recuperado el 18 de Diciembre de 2018, de http://users.dsic.upv.es/asignaturas/eui/lds/doc/masyxp.pdf

Luján, J. (s.f. de s.f de s.f.). *EDteam* . Obtenido de Patrones de diseño: https://ed.team/blog/patrones-de-diseno

Martínez, L. R. (S/N de Julio de 2015). *tesis.ipn.mx.* Obtenido de https://tesis.ipn.mx/bitstream/handle/123456789/19632/15.%20Rodolfo%20Adolfo%20Bernal%20Martinez.pdf?sequence=1&isAllowed=y

Menéndez-Barzanallana Asensio, R. (12 de noviembre de 2018). *www.um.es*. Recuperado el 07 de Enero de 2019, de Universidad de Murcia: https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-13-paginas-web-dinamicas.html

mozilla.org. (s.f. de s.f. de 2005). *MDN Web Doc Moz://a*. Recuperado el 29 de Diciembre de 2018, de https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Introducci%C3%B3n

Mozilla.org. (s.f. de s.f. de 2005). *MDN web docs moz://*. Recuperado el 30 de Enero de 2019, de developer.mozilla.org: https://developer.mozilla.org/es/docs/Web/HTML

Mozilla.org. (s.f. de s.f. de 2005). *MDN web docs moz://*. Recuperado el 30 de Enero de 2019, de developer.mozilla.org: https://developer.mozilla.org/es/docs/Glossary/CSS

NeoAttack. (s.f. de s.f. de 2018). *Concepto de API*. Recuperado el 04 de Febrero de 2019, de neoattack.com: https://neoattack.com/neowiki/api/

Orjuela Duarte, A., & Rojas C., M. (02 de Junio de 2008). *Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo.* Recuperado el 19 de Diciembre de 2018, de redalyc.org: https://www.redalyc.org/pdf/1331/133115027022.pdf

Para programadores. (2 de Marzo de 2018). *Para programadores*. Recuperado el 27 de Diciembre de 2018, de https://postparaprogramadores.com/curso-mvc/

PHP Group. (s.f de s.f de 2001). *php.net*. Recuperado el 20 de SEPTIEMBRE de 2018, de php.net: http://php.net/manual/es/preface.php

Riba, J. M. (14 de ENERO de 2008). *openaccess.uoc.edu.* Recuperado el 20 de SEPTIEMBRE de 2018, de openaccess.uoc.edu: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/876/1/00765tfc.pdf

Riehle, D. (S.f de S.f de 2000). Framework Design: A Role Modeling Approach. Zürich, Zürich, Suiza.

Riquelme, M. (14 de Mayo de 2014). ¿Qué Es Y Cómo Se Interpreta El Coeficiente De Correlación De Pearson? Recuperado el 21 de Marzo de 2019, de https://www.webyempresas.com/coeficiente-de-correlacion-de-pearson/

Rodrigo, J. A. (s.f. de Marzo de 2018). *R Pubs, Sistemas de recomendación con R*. Recuperado el 25 de Enero de 2019, de https://rpubs.com/Joaquin_AR/370301

Rodríguez Pérez, E. (s.f. de s.f. de 2017). *FRAMEWORK EN PHP PARA EL DESARROLLO Y MANTENIMIENTO DE APLICACIONES EMPRESARIALES.* Recuperado el 11 de Enero de 2019, de https://riunet.upv.es/bitstream/handle/10251/88613/RODR%C3%8DGUEZ%20-%20FRAMEWORK%20EN%20PHP%20PARA%20EL%20DESARROLLO%20Y%20MANTENIMIENTO%20DE%20APLICACIONES%20EMPRESARIALES.pdf?sequence=1

ROMERO, J. J., DAFONTE, C., GÓMEZ, Á., & PENOUSAL, F. J. (2007). *Inteligencia Artificial y Computación Avanzada.* Santiago de Compostela: Fundación Alfredo Brañas.

S. Pressman, R. (2010). *INGENIERÍA DEL SOFTWARE. UN ENFOQUE PRÁCTICO.* México: McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.

Santamaría, J., & Hernández, J. (s.f. de s.f. de s.f.). *Microsoft SQL Server.* Recuperado el 10 de Enero de 2019, de https://iessanvicente.com/colaboraciones/sqlserver.pdf

Sistemas gestores de bases de datos. (s.f de s.f de s.f). *Sistemas gestores de bases de datos.* Recuperado el 31 de Enero de 2019, de http://spain-s3-mhe-prod.s3-website-eu-west-1.amazonaws.com/bcv/guide/capitulo/8448148797.pdf

Stanford University. (1998). *Introduction to Machine Learning.* Stanford, CA 94305: Robotics Laboratory, Department of Computer Science. Recuperado el 04 de Febrero de 2019, de https://ai.stanford.edu/~nilsson/MLBOOK.pdf

UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA. (28 de Enero de 2016). *Meodología Ágil de desarrollo de software programacion extrema.* Recuperado el 04 de Febrero de 2019, de http://repositorio.unan.edu.ni/1365/1/62161.pdf

uniwebside. (s.f. de s.f. de 2006). *uniwebsidad.com*. Recuperado el 29 de Diciembre de 2018, de Introducción a CSS: https://uniwebsidad.com/libros/css/capitulo-1

ANEXOS

MANUAL DE USUARIO

Índice

MANUAL DE USUARIO 121

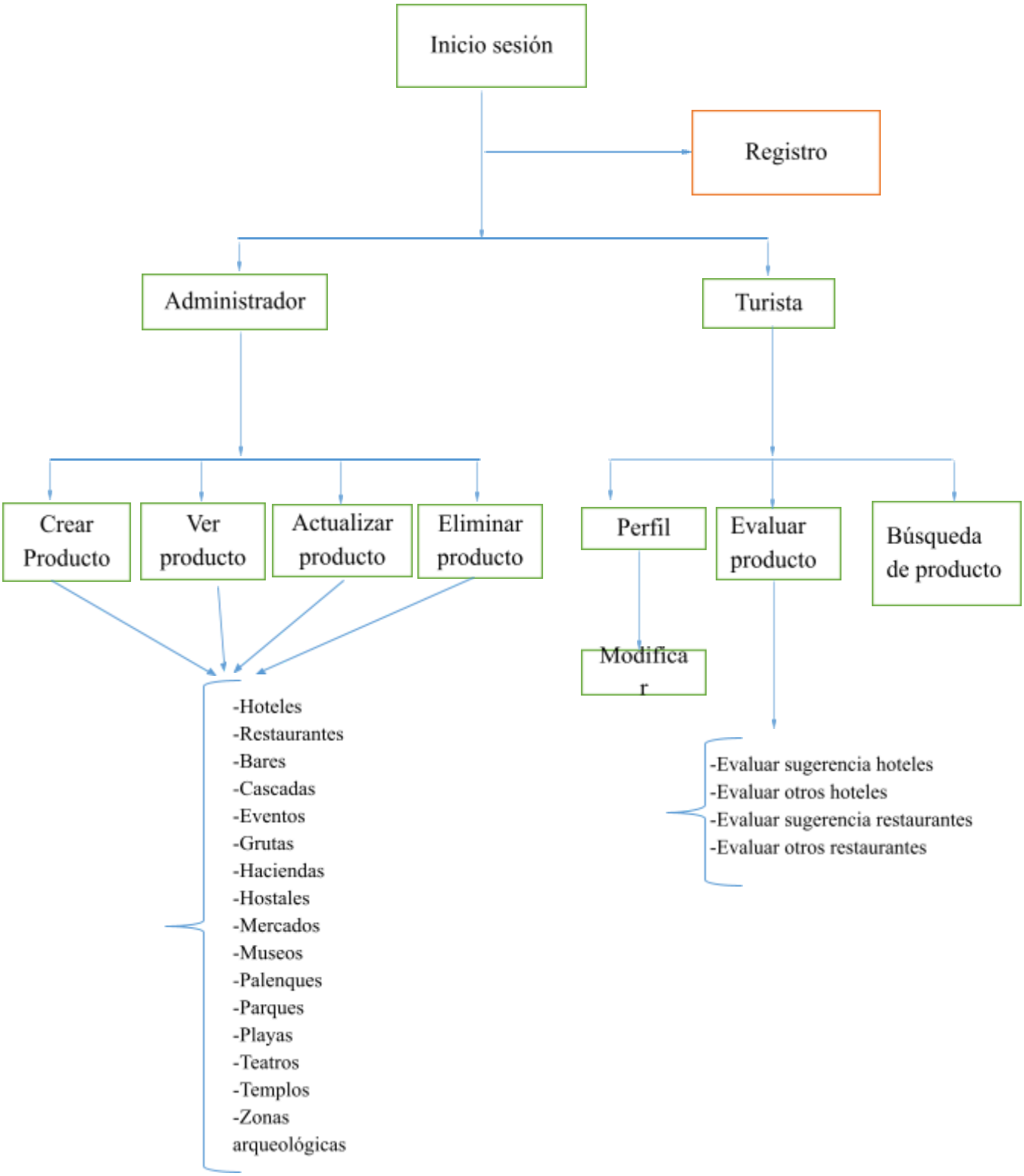
Mapa de navegación 124

1. Introducción 125
2. Confección 125
   - Nombre del Sistema: 125
   - Versión del Sistema: 125
   - Tipo de Manual: 125
   - Imagen representativa: 126
   - Fecha de Elaboración: 126
   - Área donde fue elaborado: 126
3. Requerimientos 126
4. Registro a la plataforma 127
5. Inicio de Sesión 128
6. Vista Principal Administrador 128
7. Registro de un nuevo producto Turístico 129
8. Visualización de los productos Turísticos 130
9. Actualización de datos de Productos Turísticos 131
10. Eliminación de productos turísticos 132
11. Vista principal Turista 134
12. Búsqueda de recomendaciones 135
13. Resultados de Búsqueda 137
14. Vistas en google maps 138
15. Evaluación de Restaurante/Hoteles 138

Índice de Ilustraciones

Ilustración 1 Pantalla Registro 127
Ilustración 2 Pantalla inicio sesion 128
Ilustración 3 Pantalla principal Admin. 129
Ilustración 4 Pantalla Menú de Opciones 129
Ilustración 5 Pantalla Registro Hotel 130
Ilustración 6 Pantalla Campos Llenados 130
Ilustración 7 Pantalla Visualización 131
Ilustración 8 Pantalla Listado para Edicion 131
Ilustración 9 Pantalla Edicion 132
Ilustración 10 Pantalla Listado Eliminar 132
Ilustración 11 Pantalla Datos a Eliminar 133
Ilustración 12 Pantalla Listado eliminado 133
Ilustración 13 Pantalla Principal para Turistas 134
Ilustración 14 Pantalla Busquedas 135
Ilustración 15 Pantalla Buesquedas2 136
Ilustración 16 Pantalla Resultados Busqueda 137
Ilustración 17 Pantalla visualización de resultados en maps 138
Ilustración 18 Listado a Evaluar Restaurantes 139
Ilustración 19 Listado a evaluar Hoteles 139
Ilustración 20 Pantalla Evaluación 140

Mapa de navegación



1. Introducción

El presente documento tiene como fin mostrar las diferentes funciones de la plataforma Web, y así mismo, servir como ayuda para los diferentes clientes (turistas, administradores, prestadores de servicio) para que puedan tener un mayor acercamiento a la plataforma web y aprendan a usarlo de una manera correcta.

La plataforma web tiene como funciones:

1. Manejar sesión usuario web: permite crear una cuenta con datos personales (Nombre(s), Apellidos, Correo Electrónico, Nombre Usuario, Contraseña).
2. La creación, lectura, actualización y borrado de nuevos productos turísticos (CRUD).
3. Llenar formulario para la búsqueda y recomendación de productos turísticos (hoteles y restaurantes) que vayan de acorde con el perfil del turista.
4. Puntear en google mapas las recomendaciones.
5. Evaluación de los productos turísticos para la retroalimentación del sistema y nueva generación de recomendaciones.
6. Acceso mediante distintos navegadores web.

2. Confección

Nombre del Sistema: "Plataforma Web Para la Captación, Difusión y Estudio de las preferencias, expectativas y precepciones de los turistas en Oaxaca"

Versión del Sistema: versión 1.1.

Tipo de Manual: Manual de usuario.

Imagen representativa:

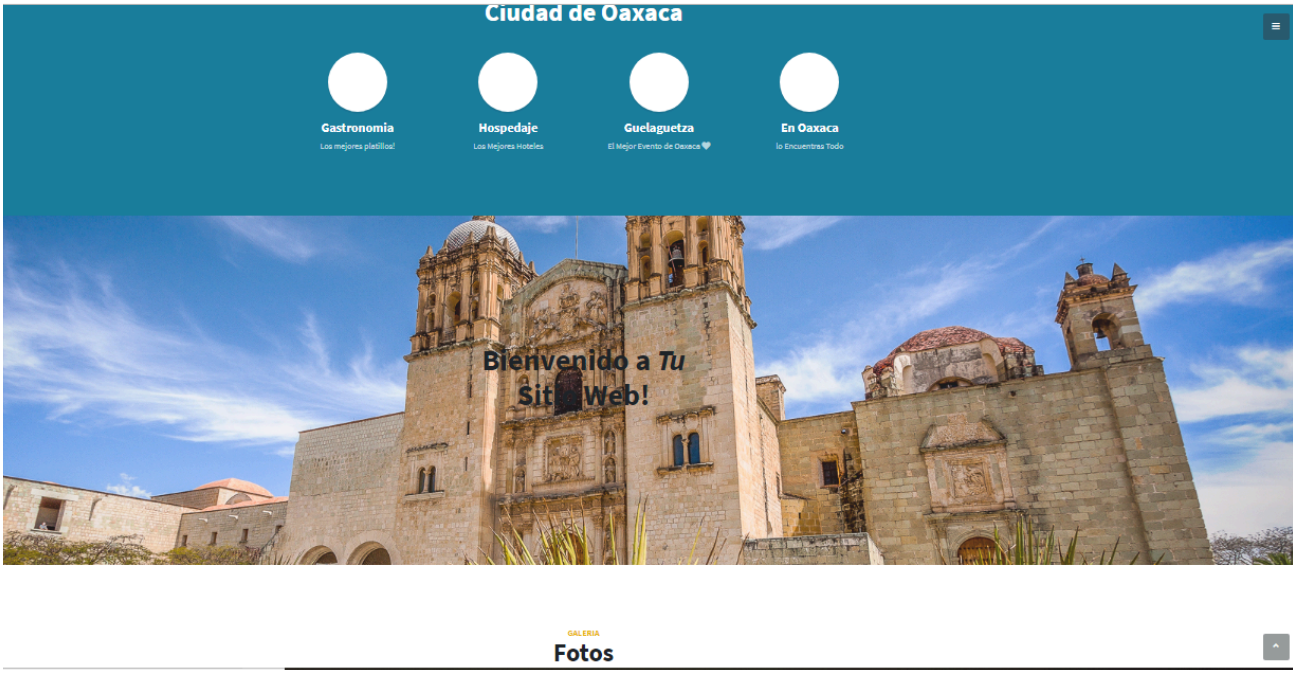


Ilustración 35 Pantalla principal

Fecha de Elaboración: 30/01/19

Área donde fue elaborado: Instituto tecnológico de Oaxaca, División de Estudios de Posgrado e Investigación.

3. Requerimientos

Los requerimientos mínimos para que la plataforma funcione correctamente, son los siguientes:

- Computadora/Dispositivo móvil con conexión a internet.
- Algún Navegador web (Se recomienda Google Chrome).

4. Registro a la plataforma

Para poder acceder a la plataforma, primero se debe de llenar el formulario de registro son campos obligatorios, una vez llenados se procede a dar clic en el botón Register.

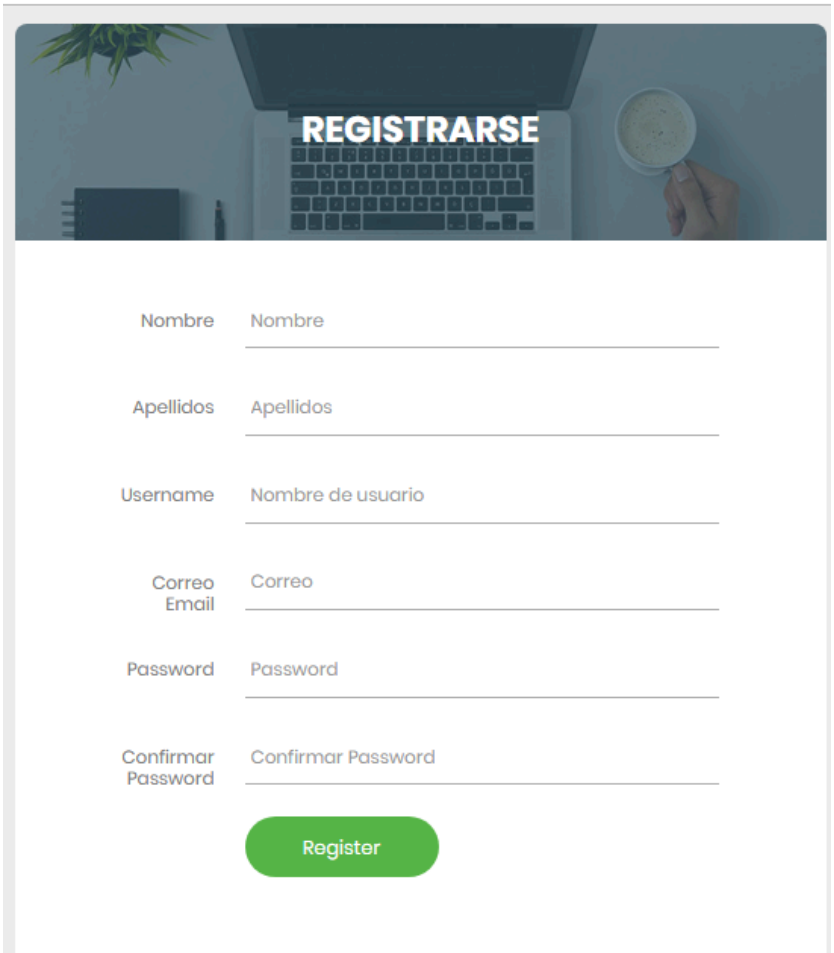


Ilustración 36 Pantalla Registro

5. Inicio de Sesión

Una vez registrados en la plataforma podremos acceder a ella, ingresando el username y password, dependiendo si es administrador o un turista podrá acceder a cierta parte de la plataforma.

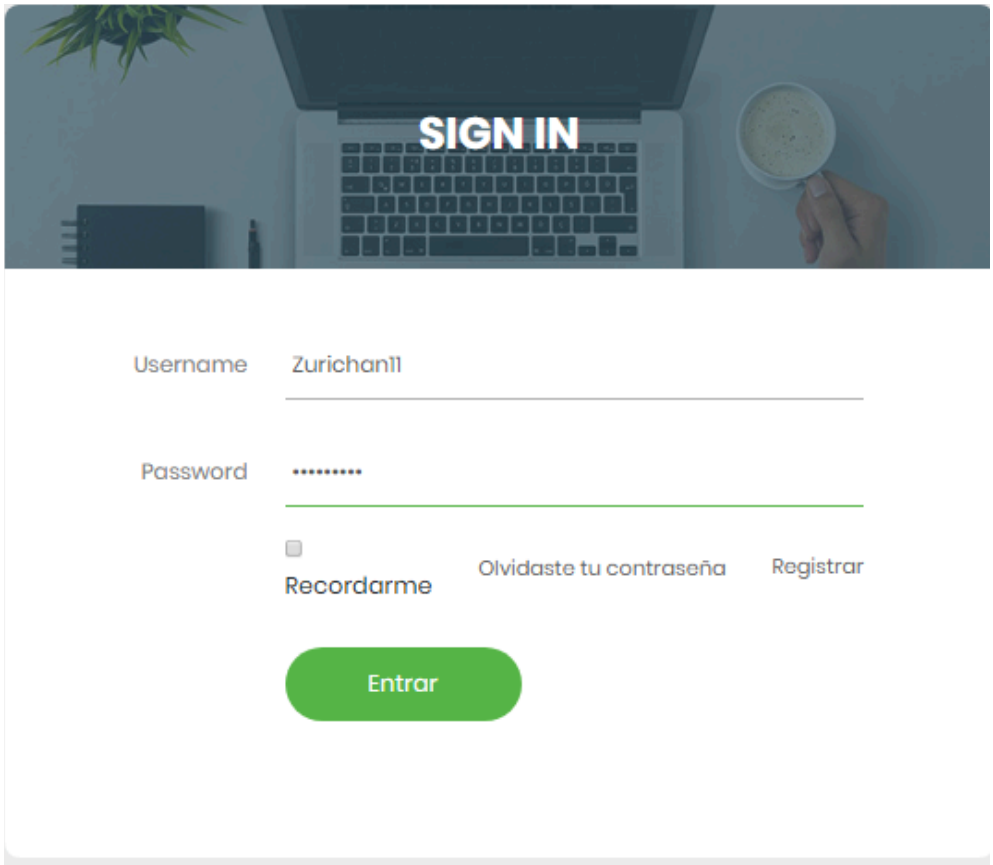


Ilustración 37 Pantalla inicio sesión

6. Vista Principal Administrador

Al haber ingresado correctamente al sistema y siendo un usuario de tipo administrador se le mostrara una pantalla en la cual podrá realizar tareas relacionadas con los productos turísticos Se mostrara un menú en la parte superior derecha en la cual desplegara un listado de nombres de productos al seleccionar alguno de ellos se mostrara opciones que podrá realizar.

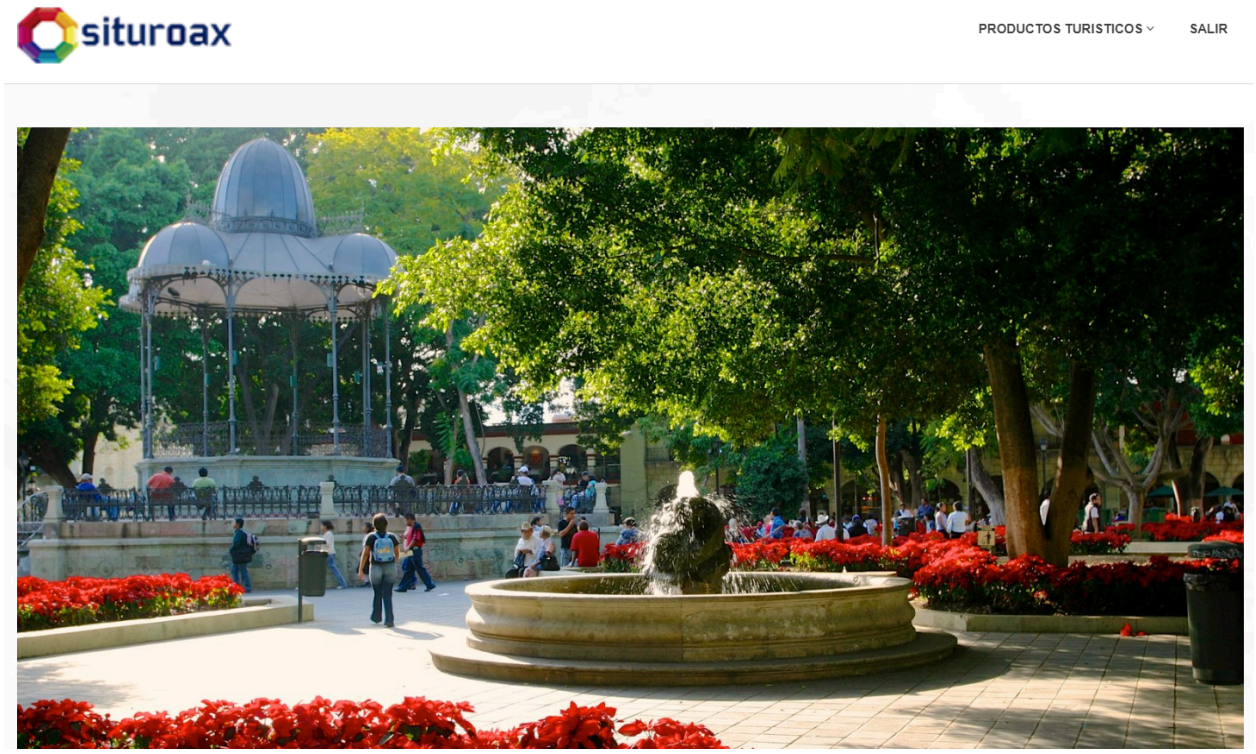


Ilustración 38 Pantalla principal Admin.

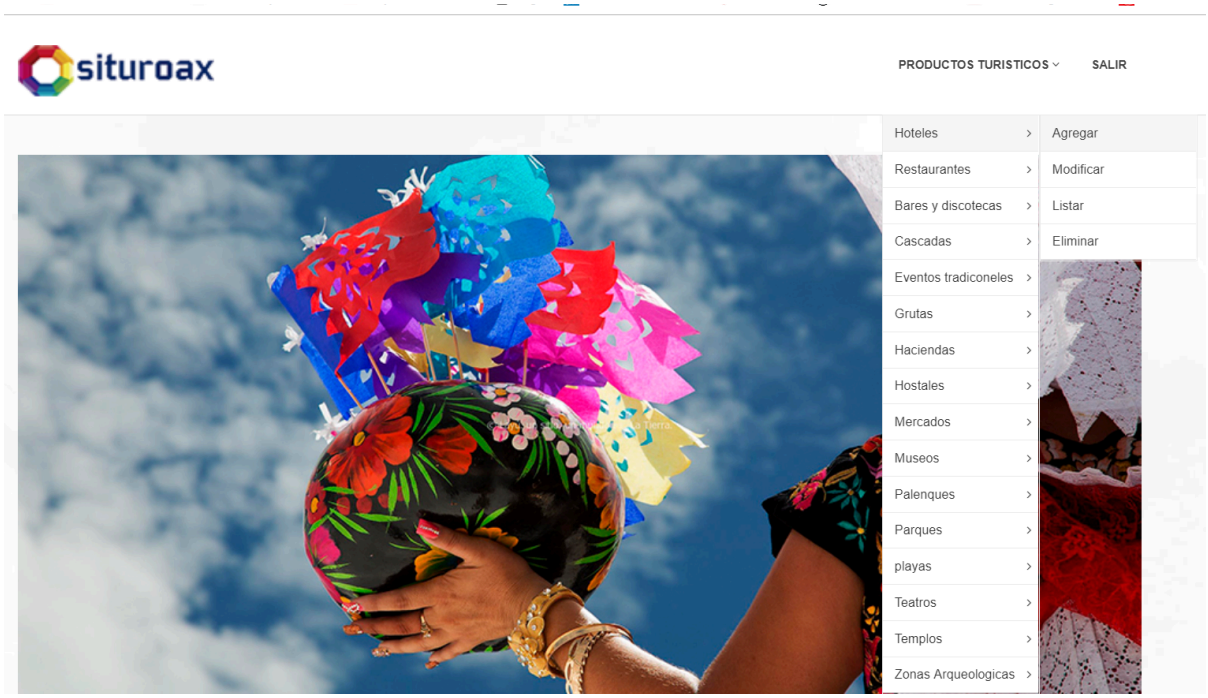


Ilustración 39 Pantalla Menú de Opciones

7. Registro de un nuevo producto Turístico

Al dar clic en la opción Agregar de alguno de los productos turísticos se mostrara el formulario para ingresar los datos (aplica para todos los productos turísticos).

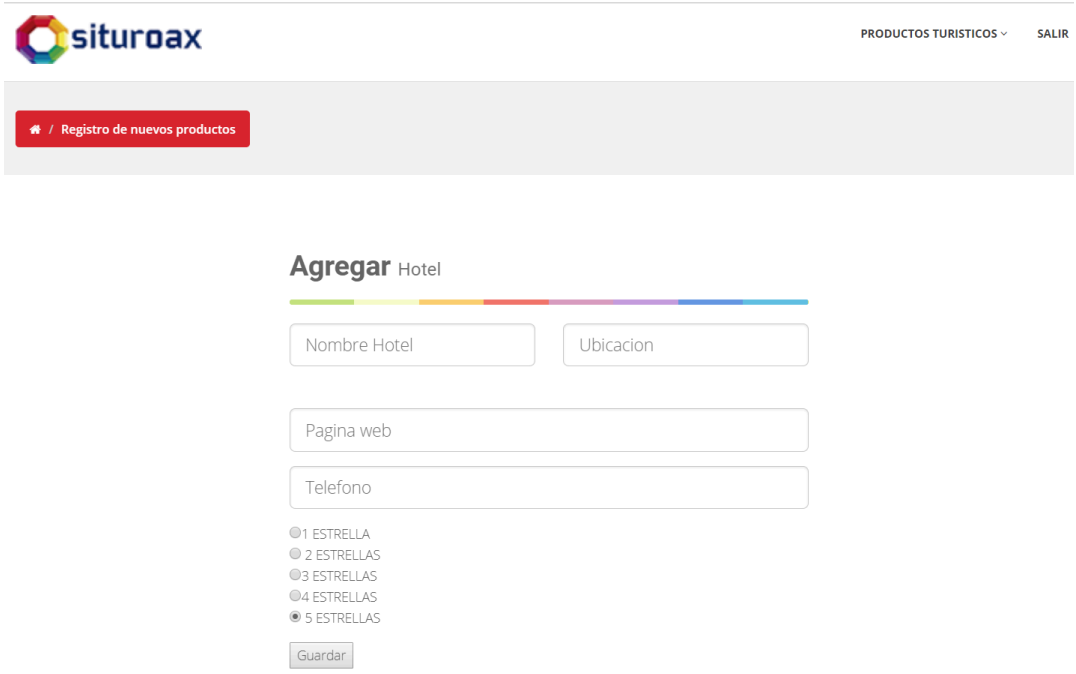


Ilustración 40 Pantalla Registro Hotel

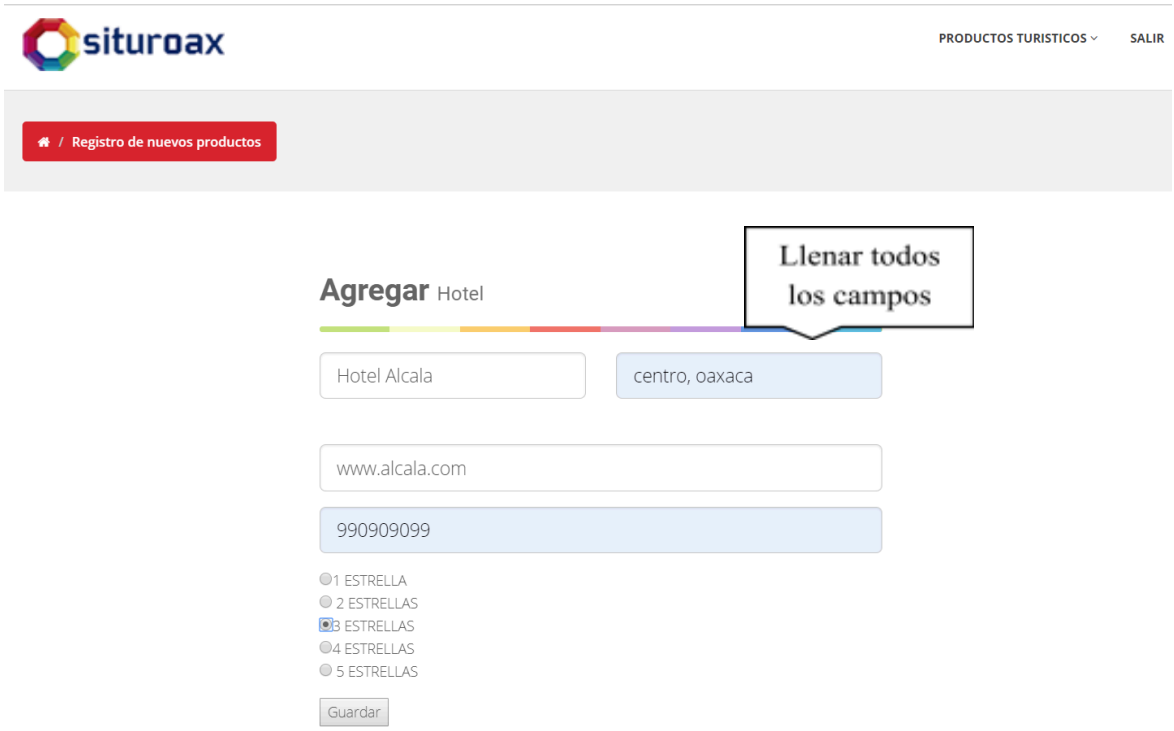
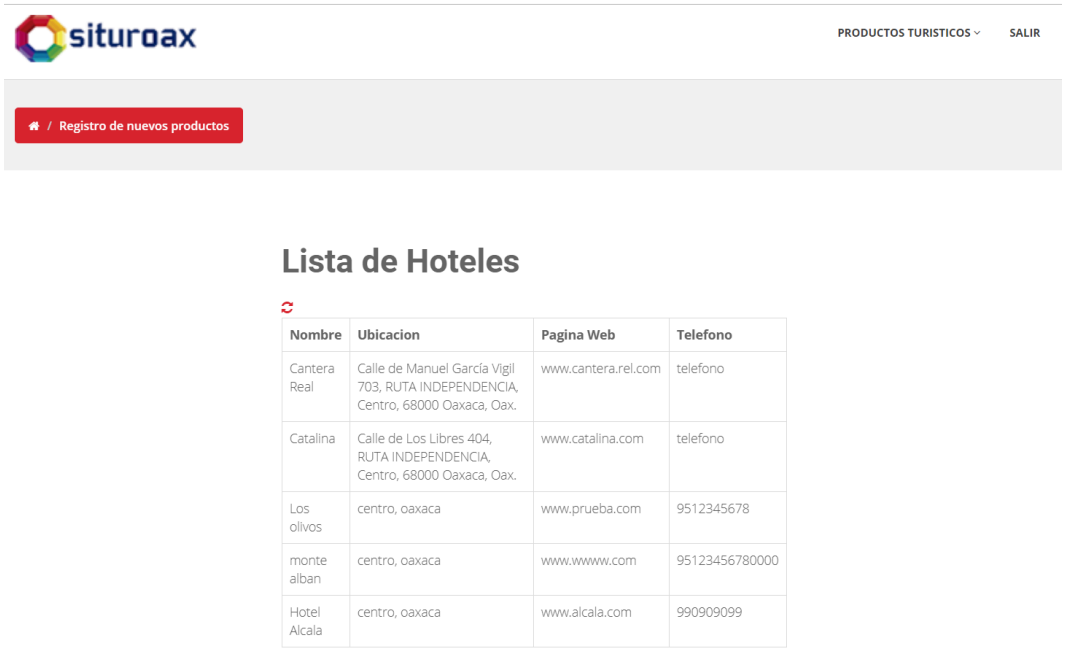


Ilustración 41 Pantalla Campos Llenados

8. Visualización de los productos Turísticos

Al dar clic en la opción visualizar algún producto turístico se mostrara una pantalla donde se listara en forma de tabla los productos de ese tipo.



Nombre	Ubicacion	Pagina Web	Telefono
Cantera Real	Calle de Manuel García Vigil 703, RUTA INDEPENDENCIA, Centro, 68000 Oaxaca, Oax.	www.cantera.rel.com	telefono
Catalina	Calle de Los Libres 404, RUTA INDEPENDENCIA, Centro, 68000 Oaxaca, Oax.	www.catalina.com	telefono
Los olivos	centro, oaxaca	www.prueba.com	9512345678
monte alban	centro, oaxaca	www.wwww.com	95123456780000
Hotel Alcala	centro, oaxaca	www.alcala.com	990909099

Ilustración 42 Pantalla Visualización

9. Actualización de datos de Productos Turísticos

Al dar clic en la opción de modificar productos turísticos se mostrara una pantalla y ahí se listaran los productos de ese tipo al dar clic en el nombre de alguno de ellos se cargara un formulario con los datos para que pueda modificar algún campo en caso de error.

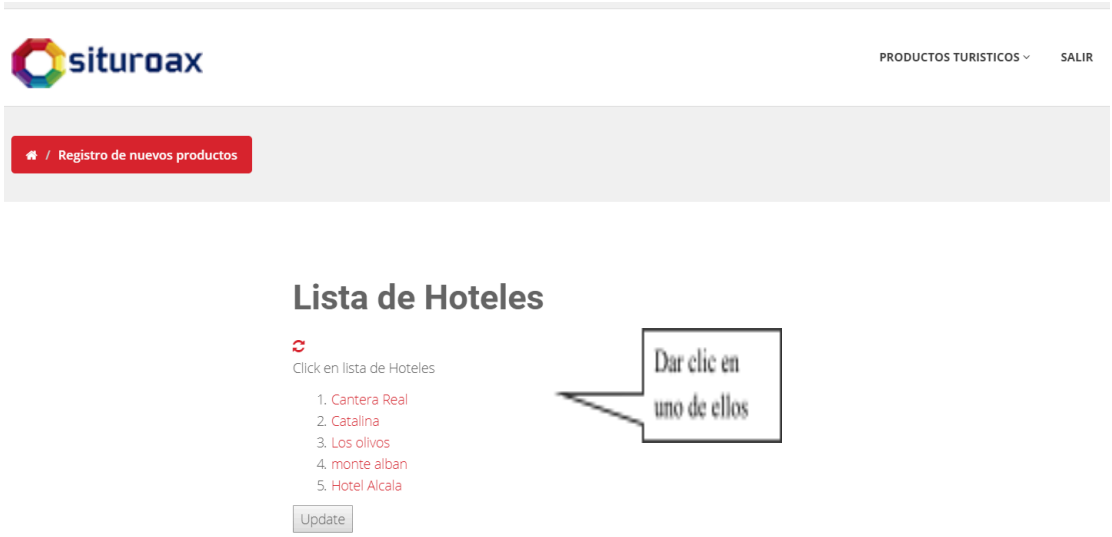


Ilustración 43 Pantalla Listado para Edición

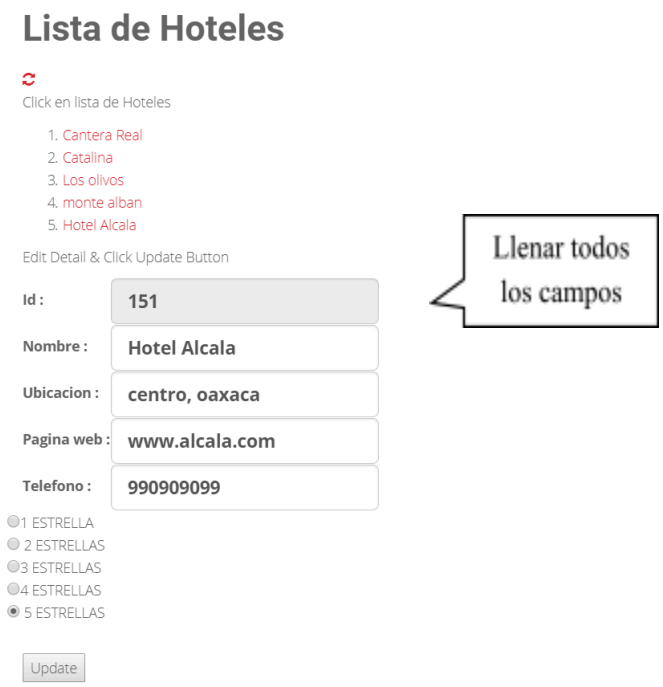


Ilustración 44 Pantalla Edicion

10. Eliminación de productos turísticos

Al dar clic en la opción de eliminar producto turístico se mostrará una pantalla y aparecerá un listado del nombre de esos productos al dar clic en alguno de ellos Se mostraran sus datos, y un botón eliminar al der cli en él se eliminara.

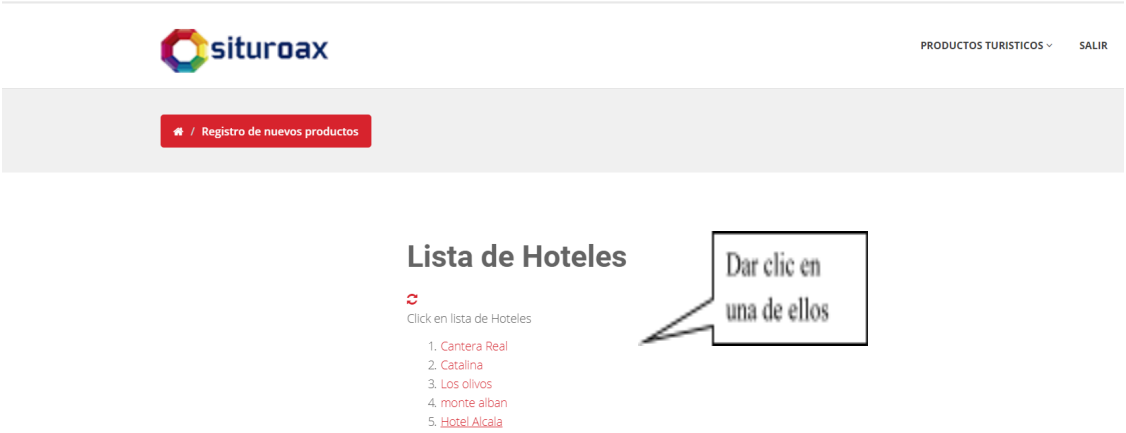


Ilustración 45 Pantalla Listado Eliminar

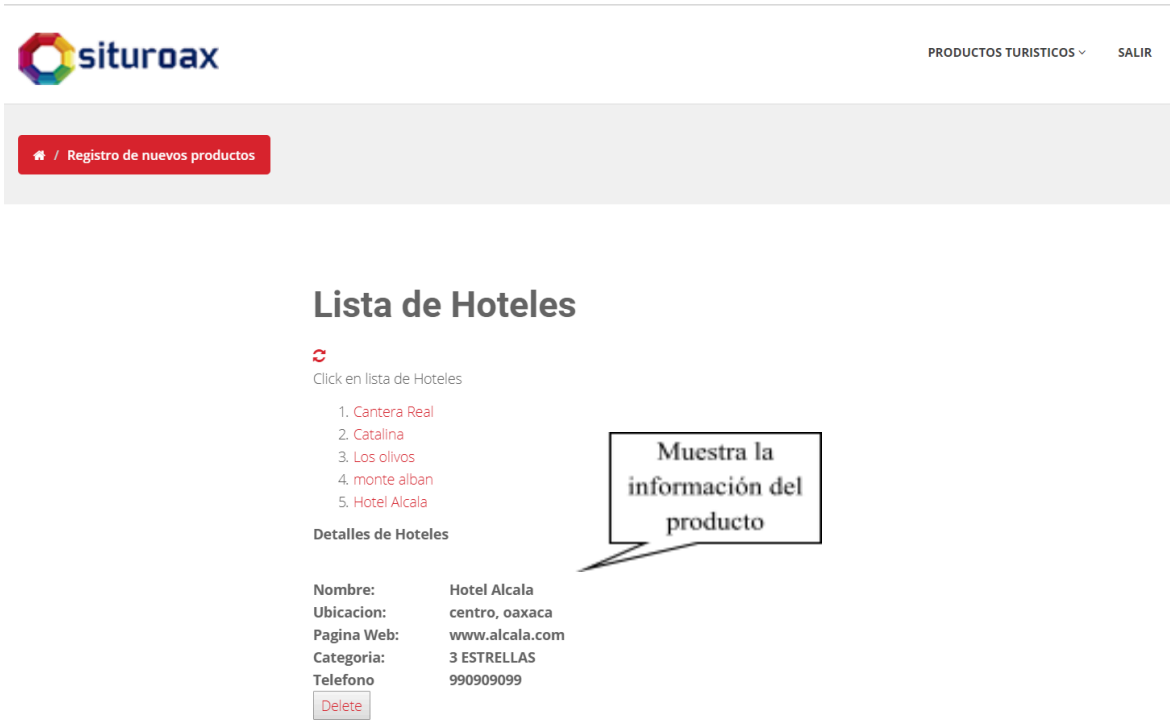


Ilustración 46 Pantalla Datos a Eliminar

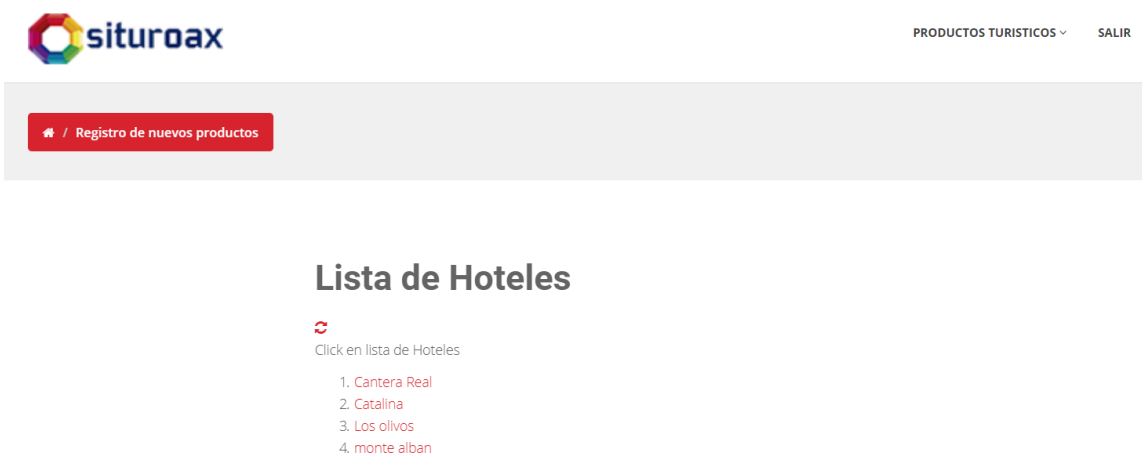


Ilustración 47 Pantalla Listado eliminado

11. Vista principal Turista

Al ingresar como usuario del tipo turista se mostrara una pantalla don la cual tendrá un menú en la parte superior derecha con la opción de perfil, productos turísticos, evaluar búsqueda.

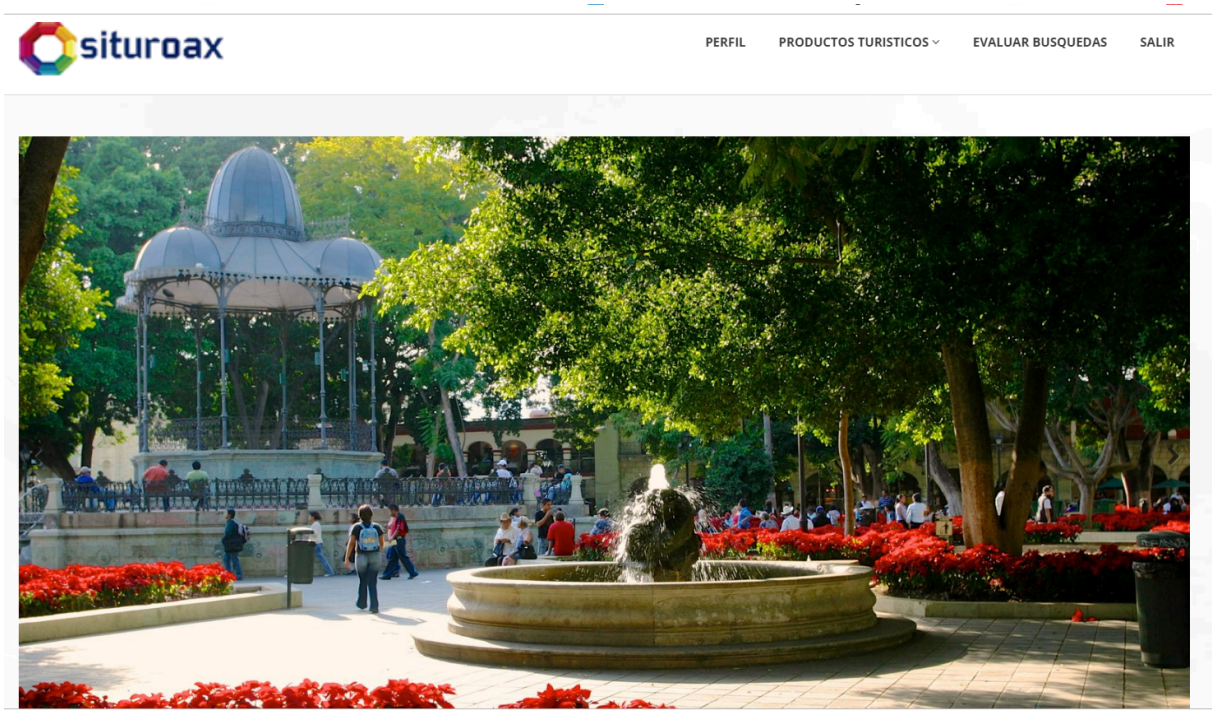


Ilustración 48 Pantalla Principal para Turistas

12. Búsqueda de recomendaciones

Al dar clic nos aparecerá una pantalla en la cual seleccionaremos algunas preferencias como turista, como el tipo de turista que nos gusta, el motivo de visita, el modo de viaje, su presupuesto, el tiempo de estancia posteriormente dar clic en realizar búsqueda.

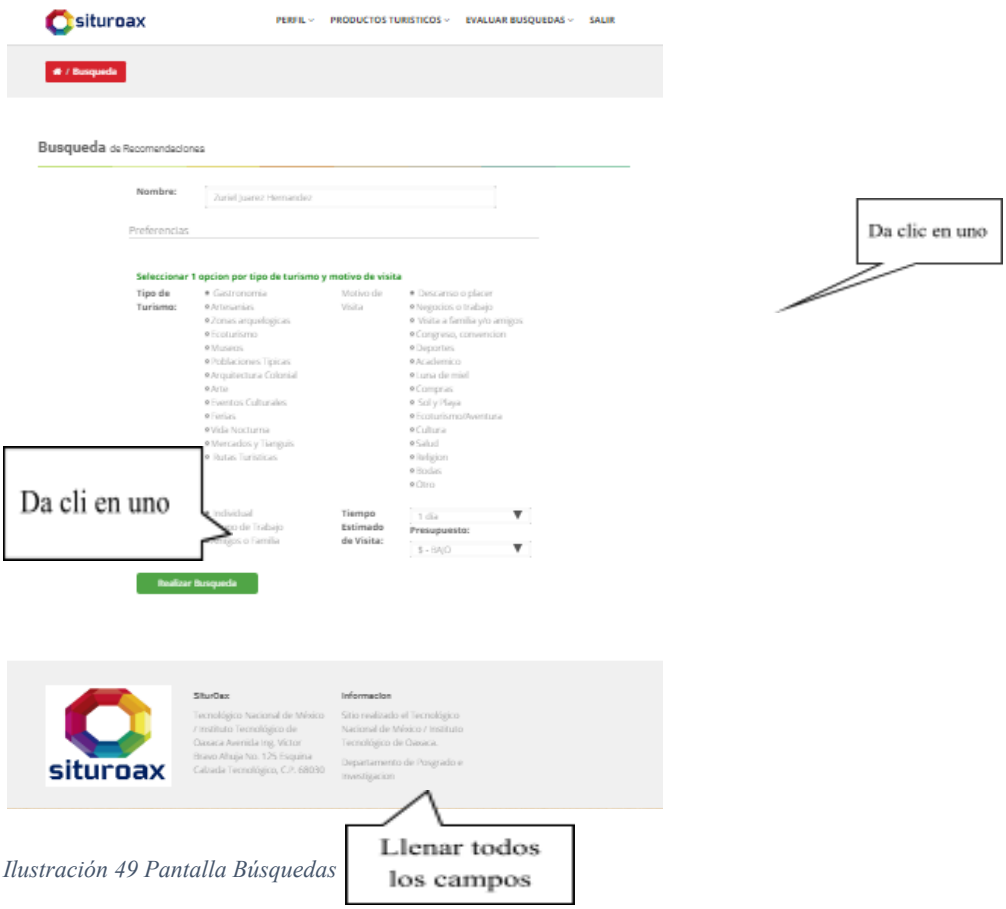


Ilustración 49 Pantalla Búsquedas

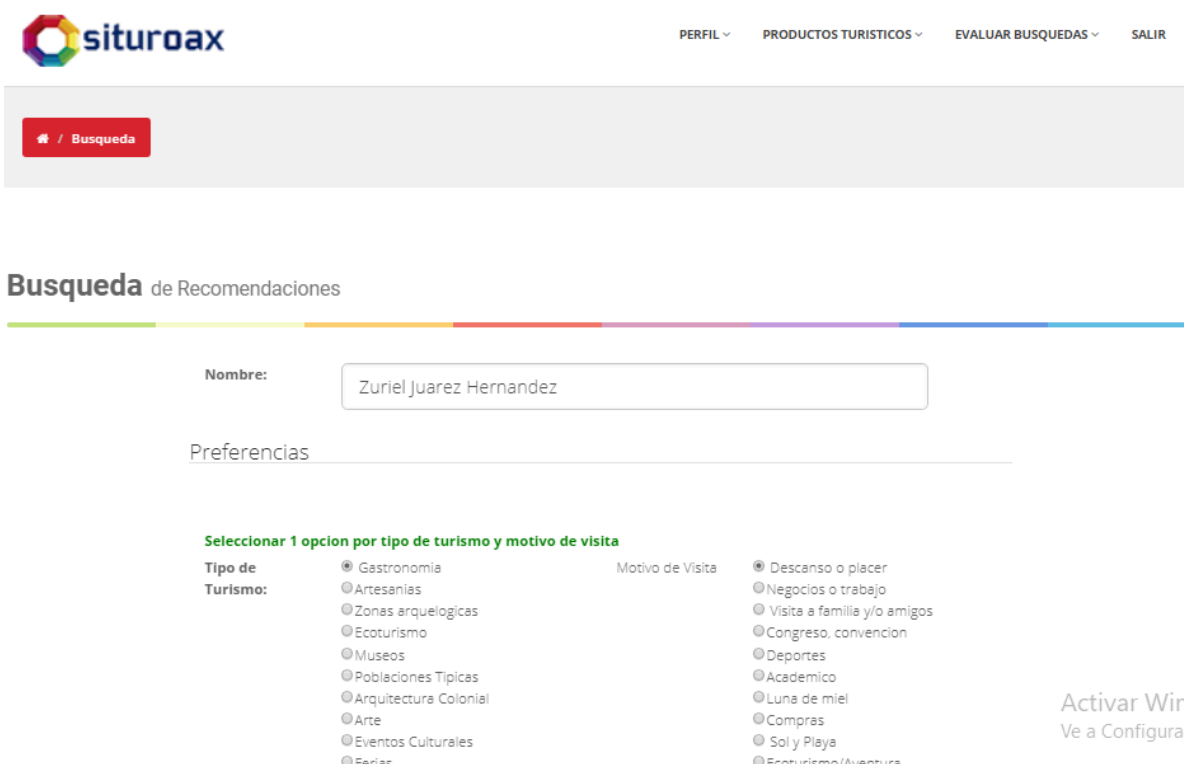


Ilustración 50 Pantalla Buesquedas2

13. Resultados de Búsqueda

Después de esperar un momento se mostrarán los resultados de la búsqueda, del lado izquierdo estarán los hoteles y restaurantes un top 5 de los que no ha visitado el turista y del lado derecho según por calificaciones de otros turistas estarán la lista de hoteles que visito y los que no ha visitado

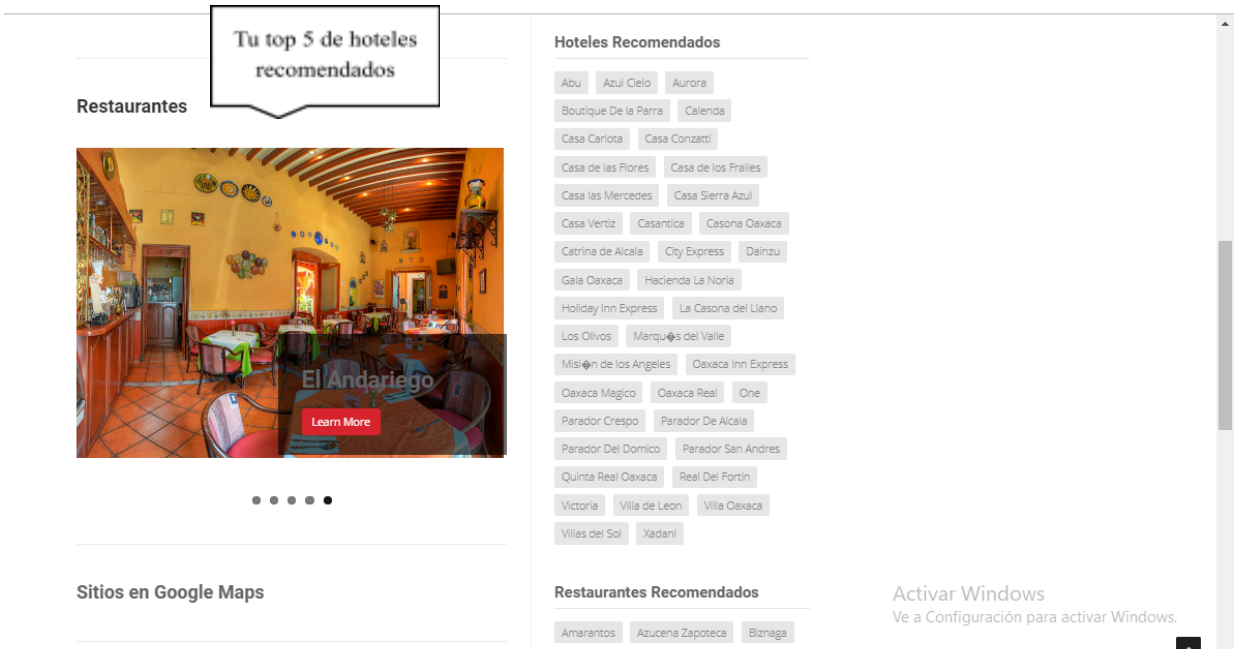


Ilustración 51 Pantalla Resultados Búsqueda

14. Vistas en google maps

Podrá dar clic en algunos de los restaurantes y o hoteles y se mostraran punteados en google maps, además de que al alejar la vista se podrán visualizara los demás hoteles y restaurantes, eso con el objetivo de que pueda tomar una decisión en base a su ubicación o si encuentra algún otro sitio de interés.

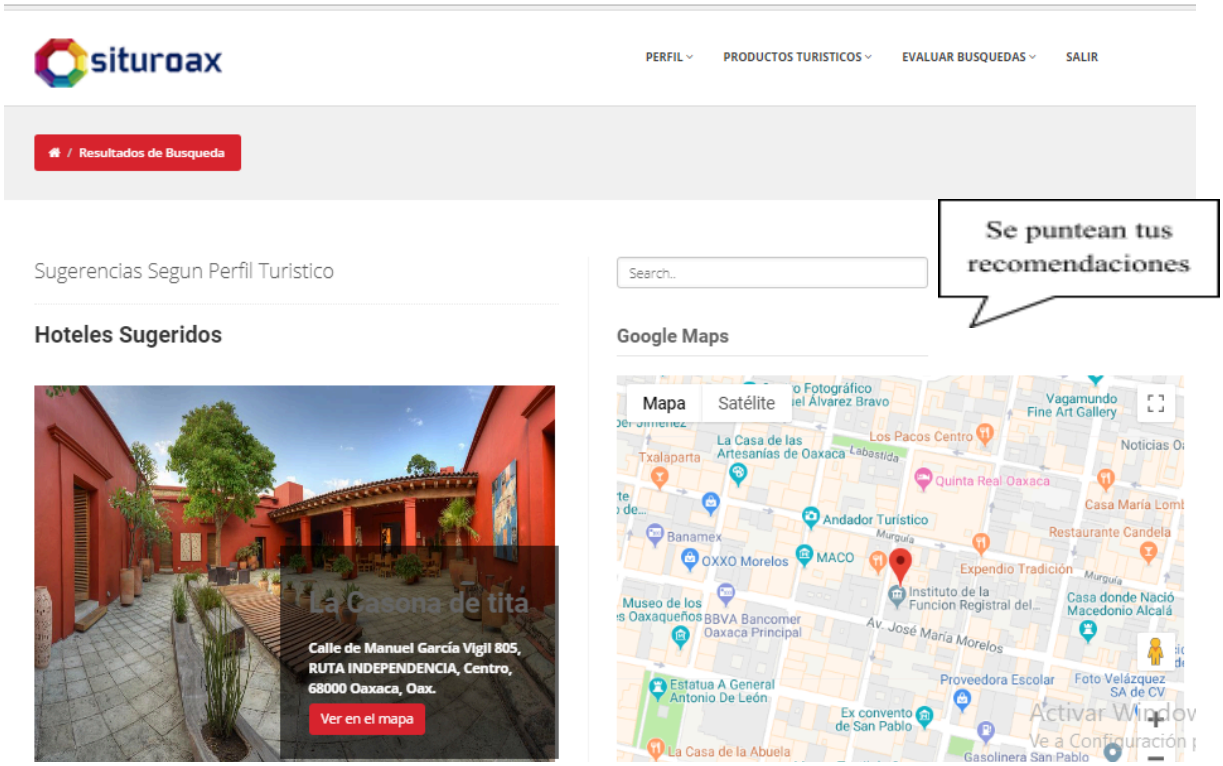


Ilustración 52 Pantalla visualización de resultados en maps

15. Evaluación de Restaurante/Hoteles

El usuario podrá evaluar los restaurantes y hoteles que se le recomendaron al dar clic en evaluar recomendación se mostrara una tabla con el id de la recomendación y la fecha que se realizó para ese turista que ingreso a la plataforma se desplegarán los nombres y al seleccionar uno aparecerá una pequeña evaluación sobre calidad del servicio una vez completado el formulario podrá guardar.

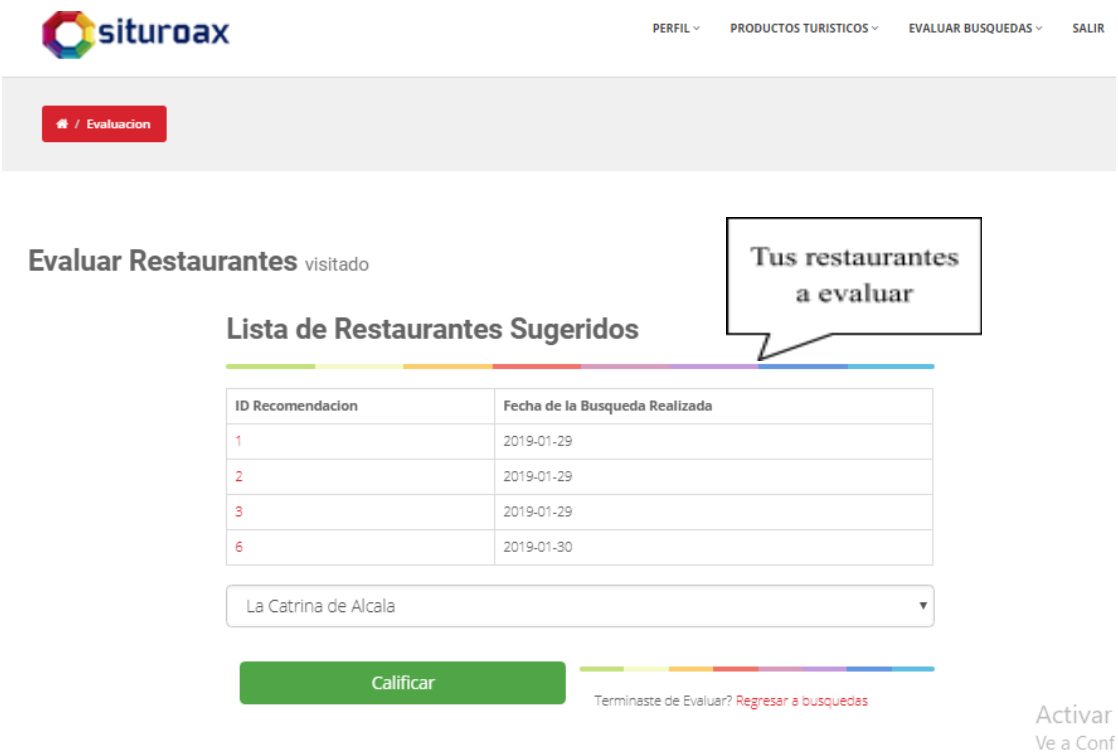


Ilustración 53 Listado a Evaluar Restaurantes

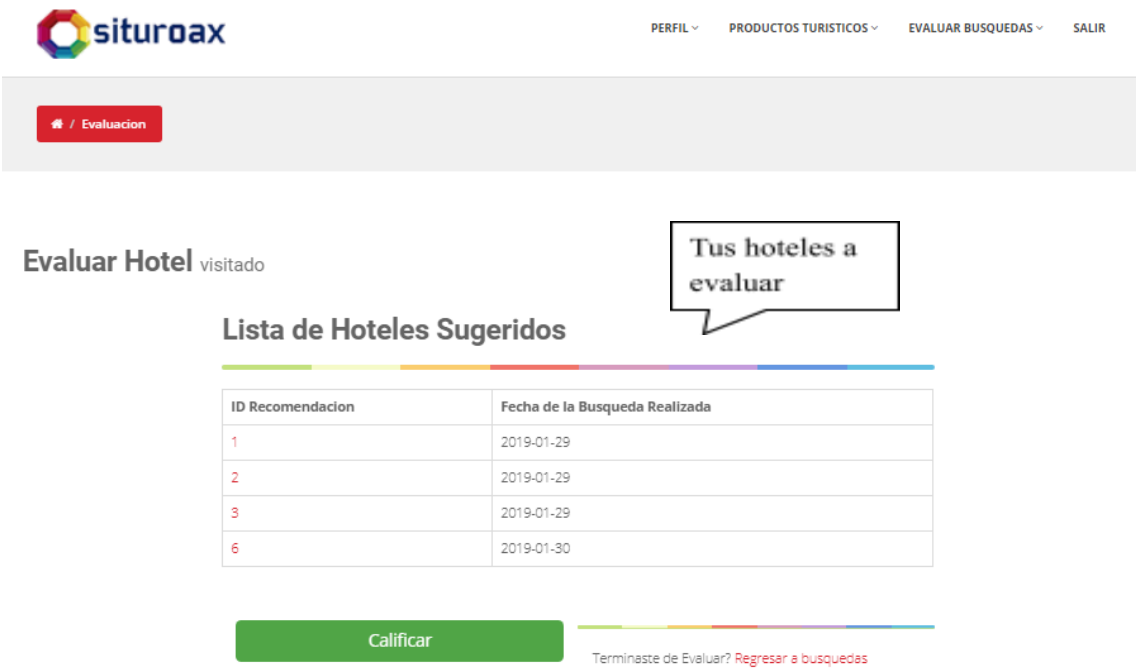


Ilustración 54 Listado a evaluar Hoteles

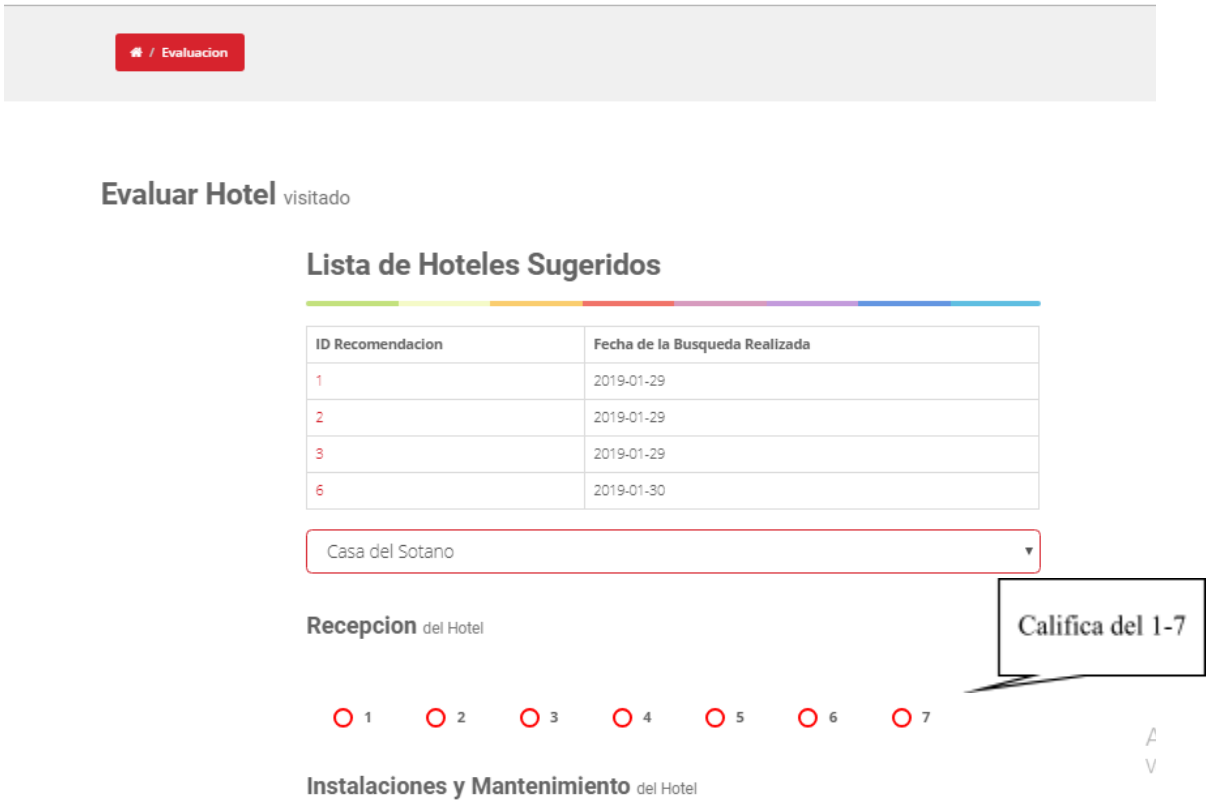


Ilustración 55 Pantalla Evaluación

2019

MANUAL TÉCNICO

[redacted]

“Plataforma Web Para la Captación, Difusión y
expectativas y precepciones de los t

GALERIA
Fotos

Contenido

MANUAL TÉCNICO 141
1. INTRODUCCIÓN 145
2. Confección 145
Nombre del Sistema: 145
Versión del Sistema: 145
Tipo de Manual: 145
Imagen representativa: 145
Fecha de Elaboración 145
Área donde fue elaborado: 145
OBJETIVO 145
ESPECÍFICOS 146
Descripción de los actores del sistema 146
REQUERIMIENTOS TÉCNICO 146
REQUERIMIENTOS MÍNIMOS DE HARDWARE 146
REQUERIMIENTOS MÍNIMOS DE SOFTWARE 146
HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO 147
PHP 147
MYSQL 147
APACHE 147
CODEIGNITER 147
INSTALACION DE WEB SERVER 148
INSTALAR Y CONFIGURAR CODEIGNITER SOBRE XAMPP 155
INSTALACION CODEIGNITER 155
CONFIGURACION CODEIGNITER 156
CONFIGURACIÓN PARA EL DISEÑO Y APLICACIÓN DE LA PLATAFORMA 158
DISEÑO Y ESTRUCURA DE LA BASE DE DATOS 158
Modulo del Sistema de Login-Diccionario de Datos 159
Módulo de Productos Turísticos-Diccionario de Datos 163

Descripción de las clases del módulo de productos turísticos 168
Controladores 168
Controlador bar 168
Controlador cascada 169
Controlador evento tradicional 169
Controlador gruta 170
Controlador hacienda 171
Controlador hostal 172
Controlador hotel 172
Controlador mercado 173
Controlador museo 174
Controlador palenque 174
Controlador parque 175
Controlador playa 176
Controlador restaurante 177
Controlador teatro 178
Controlador templo 178
Controlador zona arqueológica 179
Modelos 180
Modelo bar 180
Modelo cascada 180
Modelo evento tradicional 181
Modelo grutas 181
Modelo hacienda 182
Modelo hostal 182
Modelo hotel 183
Modelo mercado 183
Modelo museos 184
Modelo palenque 184
Modelo parques 185
Modelo playa 185

Modelo restaurantes 186
Modelo teatro 186
Modelo templo 187
Modelo zona arqueológica 187
Vistas 188
MODULO BÚSQUEDA Y RECOMENDACIÓN DE PRODUCTOS TURÍTISTICOS 189
Descripción de las clases del módulo búsquedas y recomendación de productos turísticos 193
Controladores 193

Índice Tablas

Tabla 1 Controlador bar 168
Tabla 2 Controlador cascada 169
Tabla 3 Controlador Evento Tradicional 169
Tabla 4 Controlador gruta 170
Tabla 5 Controlador hacienda 171
Tabla 6 Controlador hostal 172
Tabla 7 Controlador hotel 172
Tabla 8 Controlador mercado 173
Tabla 9 Controlador museo 174
Tabla 10 Controlador palenque 174
Tabla 11 Controlador parque 175
Tabla 12 Controlador restaurante 177
Tabla 13 Controlador teatro 178
Tabla 14 Controlador templo 178
Tabla 15 Controlador zona arqueológica 179
Tabla 16 Modelo bar 180
Tabla 17 Modelo cascada 180
Tabla 18 Modelo evento tradicional 181
Tabla 19 Modelo grutas 181
Tabla 20 Modelo hostal 182
Tabla 21 Modelo hotel 183
Tabla 22 Modelo mercado 183
Tabla 23 Modelo museos 184
Tabla 24 Modelo Palenque 184
Tabla 25 Modelo parques 185
Tabla 26 Modelo restaurantes 186
Tabla 27 Modelo teatro 186
Tabla 28 Modelo templo 187
Tabla 29 Modelo zona arqueológica 187
Tabla 30 Vistas 188

1. INTRODUCCIÓN

Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de sistemas pueda realizar la instalación de la plataforma desarrollada para el departamento de Posgrado e Investigación del Instituto Tecnológico de Oaxaca. Es importante tener en cuenta que en el presente manual se hace mención a las especificaciones mínimas de hardware y software para la correcta instalación.

2. Confección

Nombre del Sistema: “Plataforma Web Para la Captación, Difusión y Estudio de las preferencias, expectativas y precepciones de los turistas en Oaxaca”

Versión del Sistema: versión 1.1.

Tipo de Manual: Manual Técnico.

Imagen representativa:

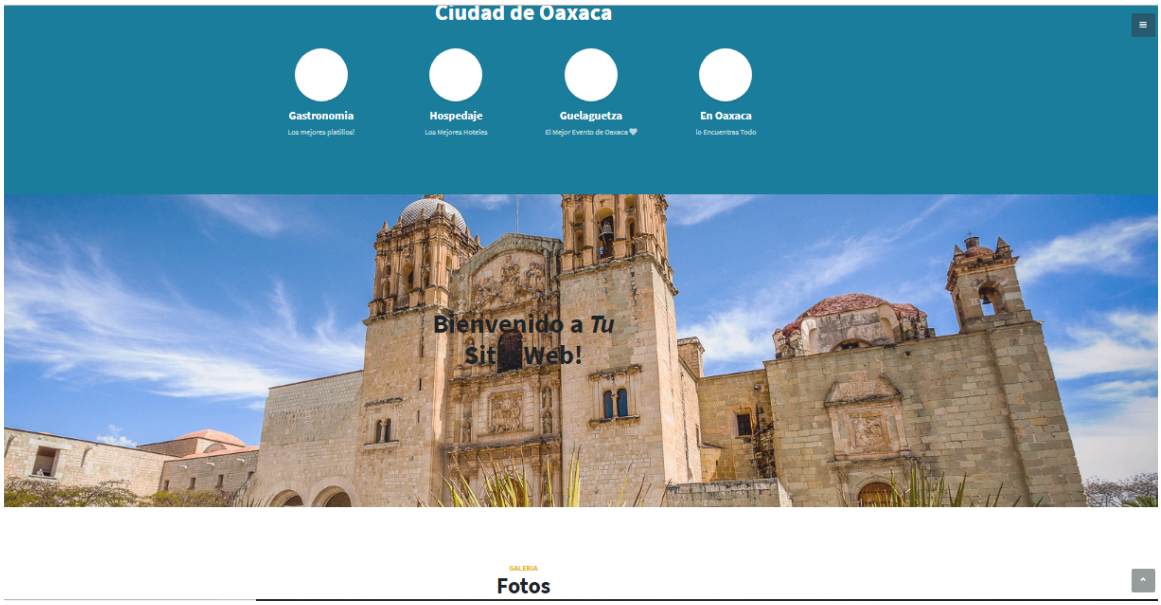


Ilustración 56 Pantalla principal

Fecha de Elaboración: 30/01/19

Área donde fue elaborado: Instituto tecnológico de Oaxaca, División de Estudios de Posgrado e Investigación.

OBJETIVO

Brindar la información necesaria para poder realizar la instalación y configuración de la plataforma.

ESPECÍFICOS

- ✔ Representar la funcionalidad técnica de la estructura, diseño y definición de la plataforma.
- ✔ Definir claramente el procedimiento de instalación de la plataforma.
- ✔ Detallar la especificación de los requerimientos de Hardware y Software necesarios para la instalación de la aplicación.
- ✔ Describir las herramientas utilizadas para el diseño y desarrollo del prototipo

Descripción de los actores del sistema

Nombre Actor	Descripción
Administrador	Se encargar de autorizar los permisos de los usuarios.
Turista	Podrá registrarse y poder realizar búsquedas de productos y evaluaciones de los mismos.

REQUERIMIENTOS TÉCNICO

REQUERIMIENTOS MÍNIMOS DE HARDWARE

- ✔ Procesador: Core i3 en adelante
- ✔ Memoria RAM: Mínimo: 1 Gigabytes (GB)
- ✔ Disco Duro: 500Gb.

REQUERIMIENTOS MÍNIMOS DE SOFTWARE

- ✔ Privilegios de administrador
- ✔ Sistema Operativo: Windows 8/Windows 10

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

PHP

Es un Lenguaje de Programación para trabajar páginas WEB ofreciendo la ventaja de mezclarse con HTML. Las ejecuciones son realizadas en el Servidor y el cliente es el encargado de recibir los resultados de la ejecución. Si el cliente realiza una petición, se ejecuta el intérprete de PHP y se genera el contenido de manera dinámica. Permite conexión con varios tipos de Bases de Datos como: MySql, Oracle, PostgreSQL, SQL Server, etc. permitiendo aplicaciones robustas sobre la WEB. Este lenguaje de programación puede ser ejecutado en la gran mayoría de sistemas operacionales y puede interactuar con Servidores WEB populares

MYSQL

Es un manejador de Bases de Datos, el cual permite múltiples hilos y múltiples usuarios, fue desarrollado como software libre. Aunque se puede usar sobre varias plataformas es muy utilizado sobre LINUX. Es libre para uso en Servidores WEB. Ofrece ventajas tales como fácil adaptación a diferentes entornos de desarrollo, Interacción con Lenguajes de Programación como PHP, Java Script y fácil Integración con distintos sistemas operativos

APACHE

Es un Servidor WEB desarrollado por el grupo Apache. Su código fuente se puede distribuir y utilizar de forma libre. Está disponible para diferentes plataformas de Sistemas Operativos entre otros Windows, Linux, Mac y NetWare. Ofrece ventajas tales como independencia de plataforma, haciendo posible el cambio de plataforma en cualquier momento; creación de contenidos dinámicos, permitiendo crear sitios mediante lenguajes PHP.
Además de ser libre su soporte técnico es accesible ya que existe una comunidad que está disponible en foros, canales IRC y servidores de noticias, donde hay gran cantidad de usuarios disponibles para cuando surge algún problema.

CODEIGNITER

CodeIgniter se basa libremente en el popular patrón de desarrollo modelo-vista-controlador (MVC). Si bien las clases de controladores son una parte necesaria del desarrollo en CodeIgniter, los modelos y las vistas son opcionales.

Codeigniter también se puede modificar para usar el controlador de vista de modelo jerárquico (HMVC) que permite a los desarrolladores mantener la agrupación modular de controlador, modelos y vista organizados en un formato de subdirectorio.
CodeIgniter es más conocido por su velocidad en comparación con otros marcos de trabajo de PHP En una versión crítica de los marcos de PHP en general

INSTALACION DE WEB SERVER

La instalación de XAMPP en Windows es realmente fácil y sin problemas en Windows 7 a 10.
Primero que todo, ir al sitio oficial de XAMPP:
https://www.apachefriends.org/es/index.html

Figura 1 Descarga xampp

En la pantalla de Descargar, se elige el servidor XAMPP a instalar. La versión de PHP a utilizar es 5.6,
Es recomendable y se sugiere la instalación de PHP 5.6, pues se trabaja con el FRAMEWORK CODEIGNITER, que trabaja bajo esta misma versión

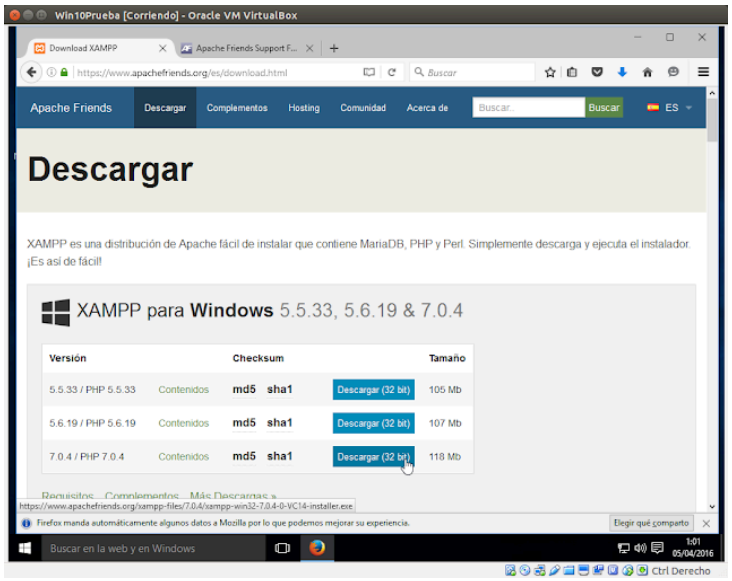


Figura 2 Versión PHP

Una vez descargado el XAMPP, se Ejecuta como administrador.
El programa de instalación arrancará y mostrará la siguiente pantalla, donde el Windows pregunta si queremos que el instalador de XAMPP haga cambios en el PC. Pulsamos sobre Si.

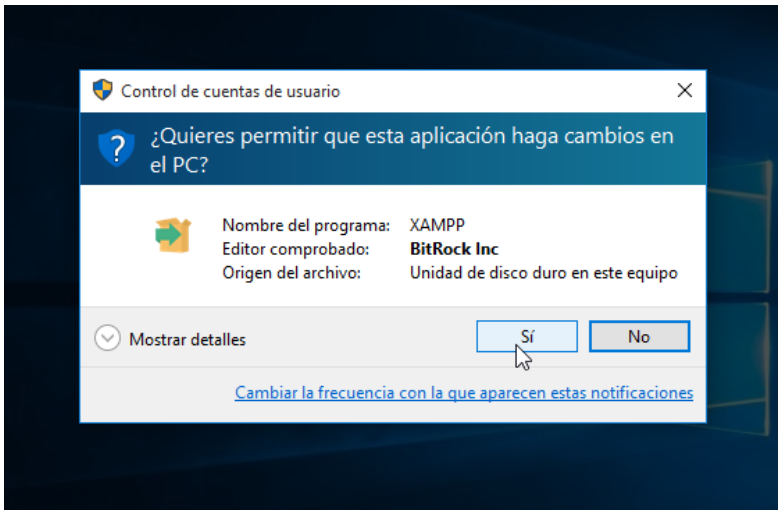


Figura 3 Ejecutar como administrador

A continuación saldrá la siguiente pantalla. El Control de cuentas de usuario (UAC) es un componente de seguridad de Windows y que permite a los usuarios realizar tareas comunes como no administradores y como administradores pero sin tener que cambiar de usuario, cerrar sesión ni utilizar Ejecutar como, pero puede que de errores en algunas ocasiones si no se ejecuta el programa como administrador.
Si eres el único usuario y además administrador de tu PC, puedes desactivar el UAC, pero no te lo aconsejo, ya que sirve de recordatorio del cuidado que debes tener al instalar programas en tu equipo.

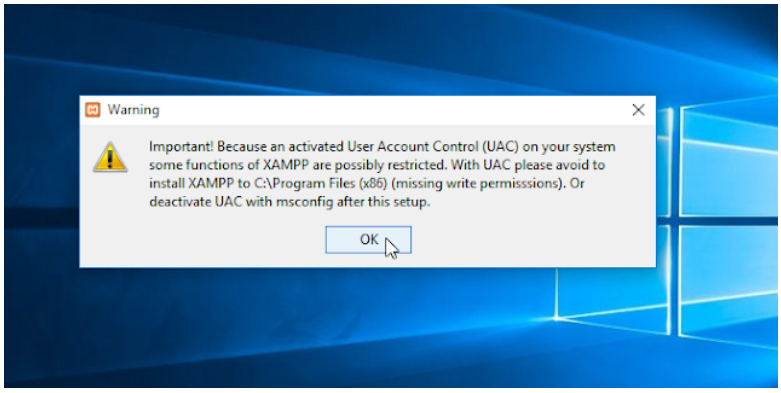


Figura 4 Advertensia

Inmediatamente se abre la siguiente ventana y pulsamos en la opción Next.

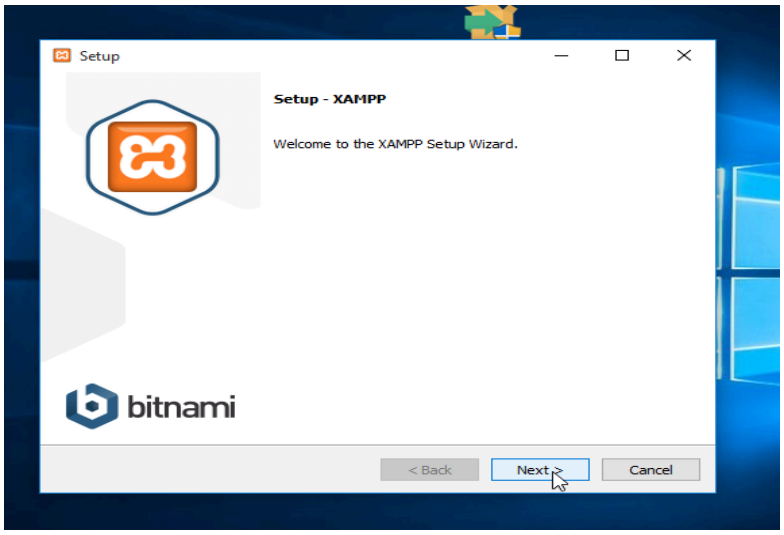


Figura 5. Instalación

En la siguiente pantalla que se presenta se elige que vamos a instalar y que no en nuestro servidor. Podemos escoger todas las opciones como en este caso, Pulsamos sobre el botón Next.

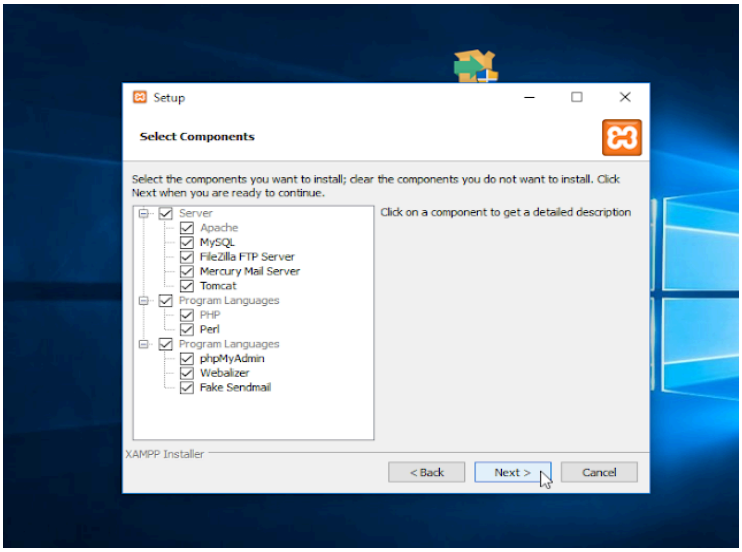
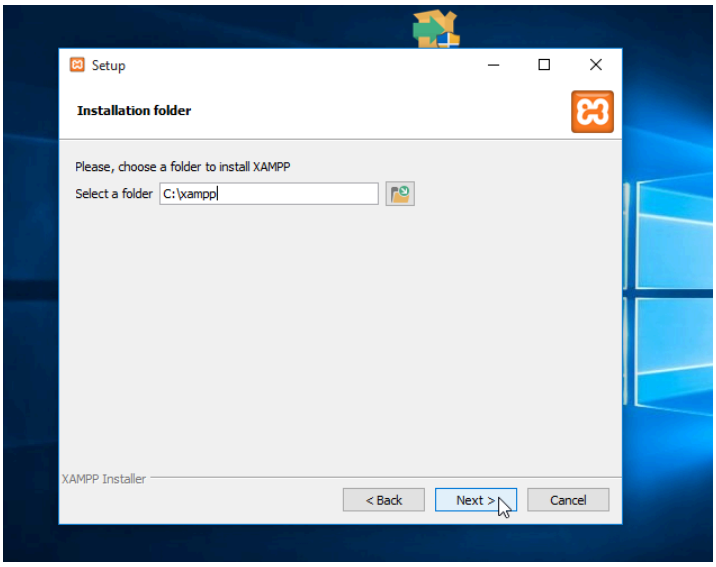


Figura 6 configuraciones

A continuación la siguiente ventana nos sugiere donde instala el XAMPP. En este caso lo instalaremos donde lo sugiere el programa de instalación. Pulsamos Next.



La siguiente pantalla simplemente nos informa de que el programa de instalación está preparado para instalar el XAMPP en el PC. Si queremos cambiar algo podemos hacerlo ahora y pulsar el botón Back para hacer los cambios, sino para empezar a instalar XAMPP con la configuración que tenemos solo pulsamos sobre Next.

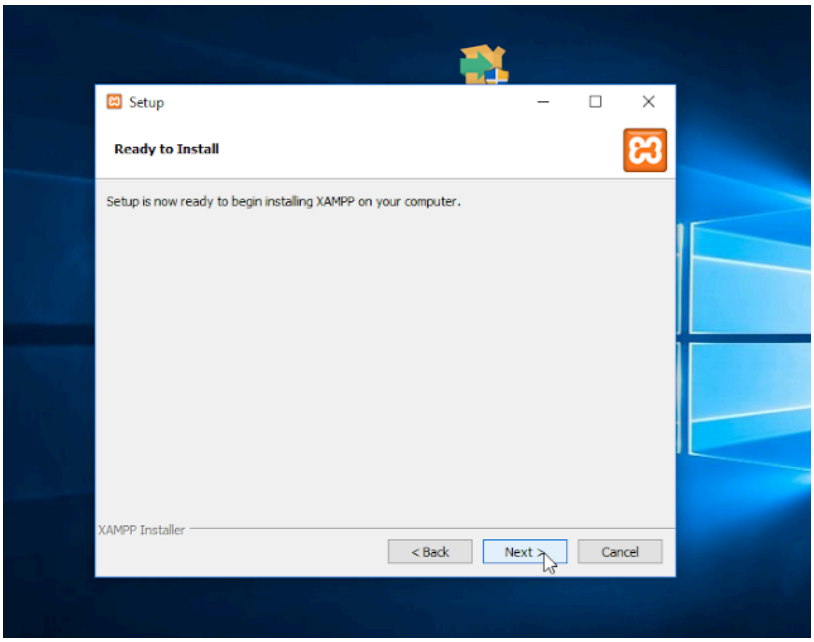


Figura 7 instalación next

Nos aparece la siguiente pantalla que nos informa el progreso de la instalación.

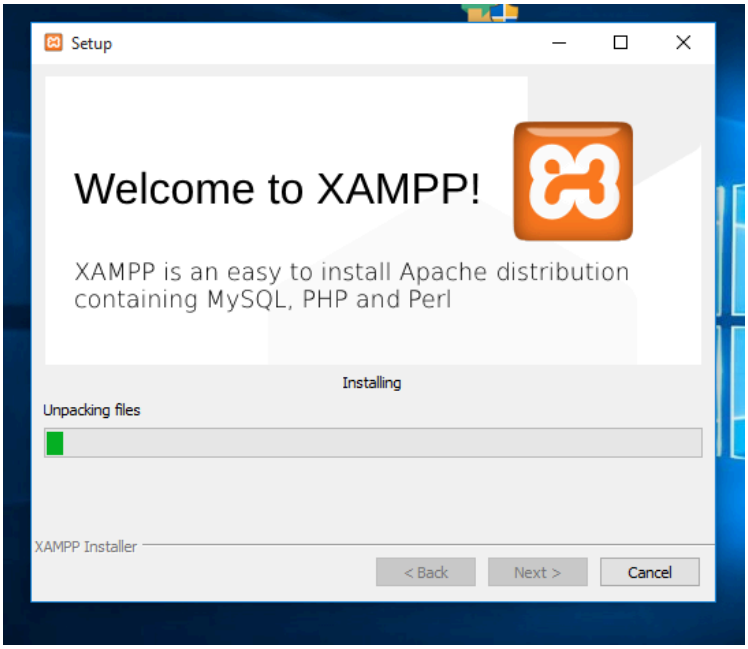
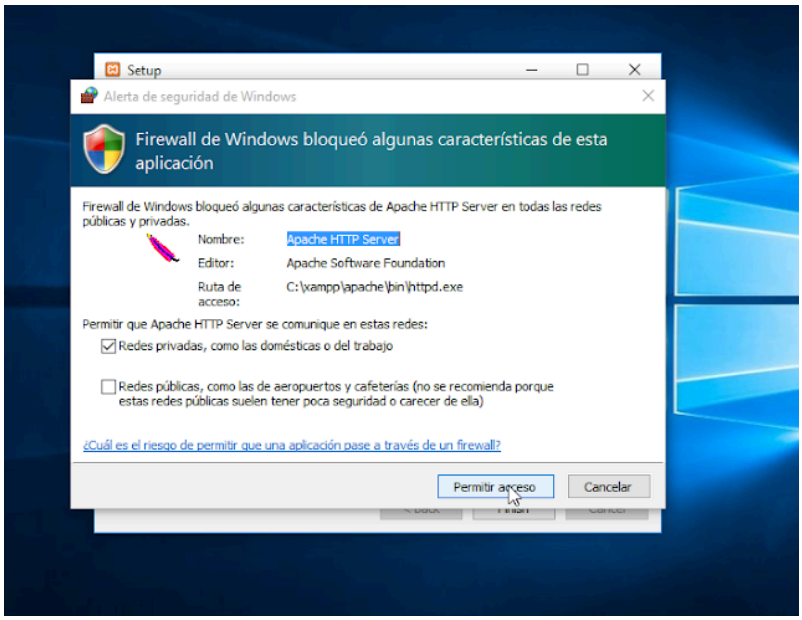


Figura 8 instalando

En un momento dado el Firewall de Windows nos salta, diciendo que ha bloqueado algunas características del servidor Apache de XAMPP para las redes públicas y privadas. Aquí simplemente debemos escoger as opciones que vamos a utilizar. Si es en un entorno Local, o sea, una Red interna dejaremos marcada la casilla Redes privadas, como domésticas o del trabajo. Esta es la que pone por defecto al Firewall. Pero si el uso del servidor va a ser público,

debemos marcas la casilla Redes públicas. El hacer esto expone el equipo a ataques, con lo cual deberemos protegerlo adecuadamente. Podemos optar por ponerlo detrás de proxy, pero eso es otro tema. Dejemos marcada la casilla Redes privadas, para esta instalación. Pulsamos sobre el botón Permitir acceso. Ojo, el programa de instalación, no se detiene en esta pregunta, el seguirá instalando, ya que el que está preguntando es el Windows.



}

Figura 9 permisos

Finalmente aparece la siguiente pantalla que indica la finalización de la instalación de XAMPP. Si no queremos que el Control Panel del XAMPP arranque ahora, dejamos sin marcar la casilla Do you want to start the Control Panel now? Aquí lo marcaremos para iniciar de una vez el XAMPP, ya que hay que hacer unas configuraciones más luego de instalar XAMPP. Pulsamos el botón Finish.

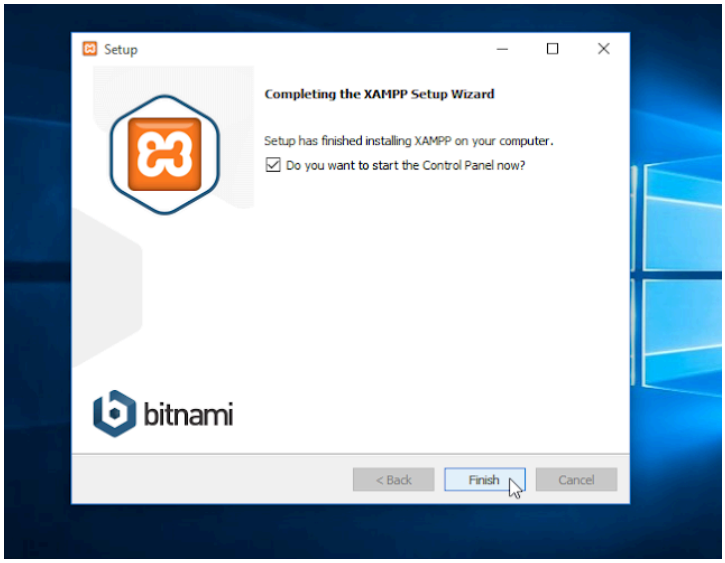


Figura 10 finalización

Si hemos marcado la casilla de arrancar el Control Panel ahora, aparecerá la siguiente pantalla, donde escogeremos el idioma que vamos a utilizar. Marcamos el que mejor entendamos, en nuestro caso será el inglés y pulsamos el botón **Save**.

Figura 11. Idioma

Nos aparecerá la siguiente pantalla, donde arrancaremos el servidor pulsando el botón Start que está a la derecha del Módulo Apache.

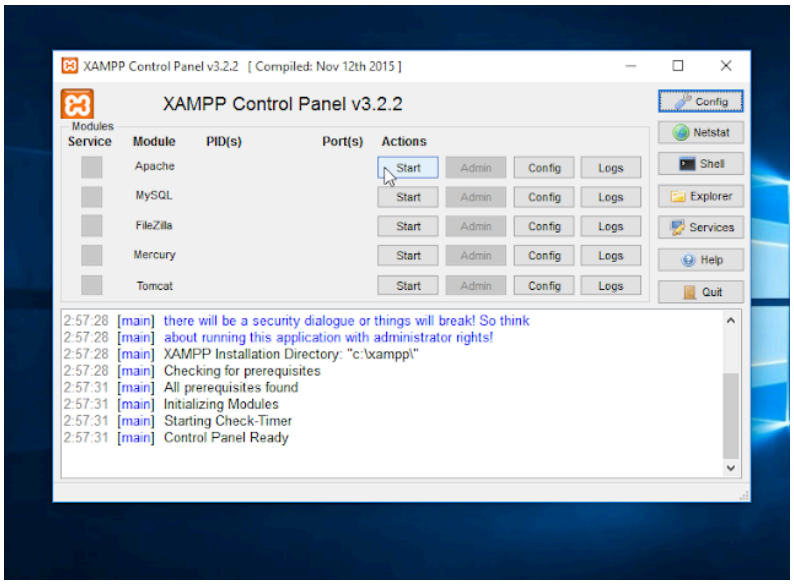


Figura 12. Panel

Luego procedemos hacerlo mismo con el Modulo del MySQL.

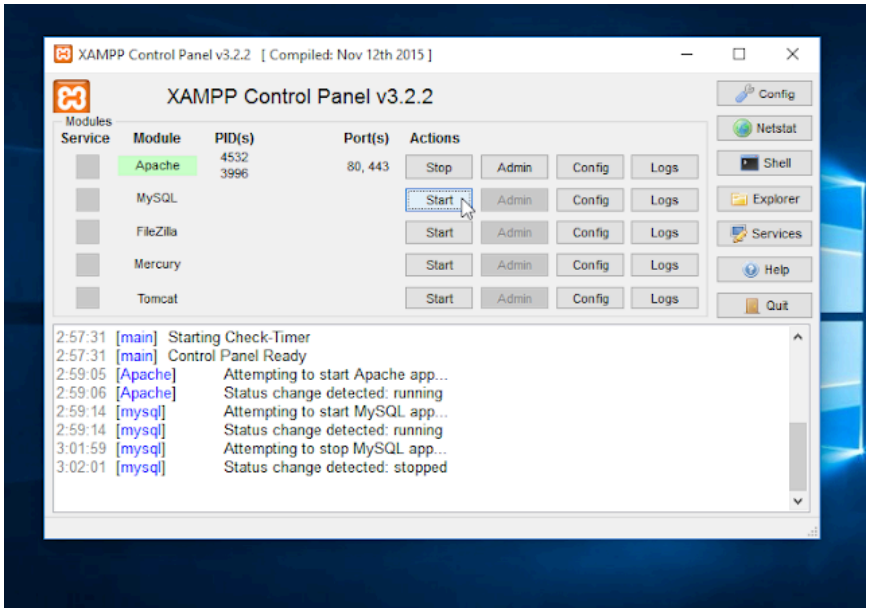


Figura 13. Panel iniciando

Al arranque del servidor de base de Datos MySQL, el Firewall de Windows vuelve a pedirnos que redes vamos a permitir que se comuniquen con él. Escogeremos la opción que hemos escogido antes para el servidor Apache si es el caso. Pero en nuestro caso dejaremos permitir conectarse a las Redes privadas. Co lo cual dejamos marcadas Redes privadas y pulsamos sobre el botón Permitir acceso.

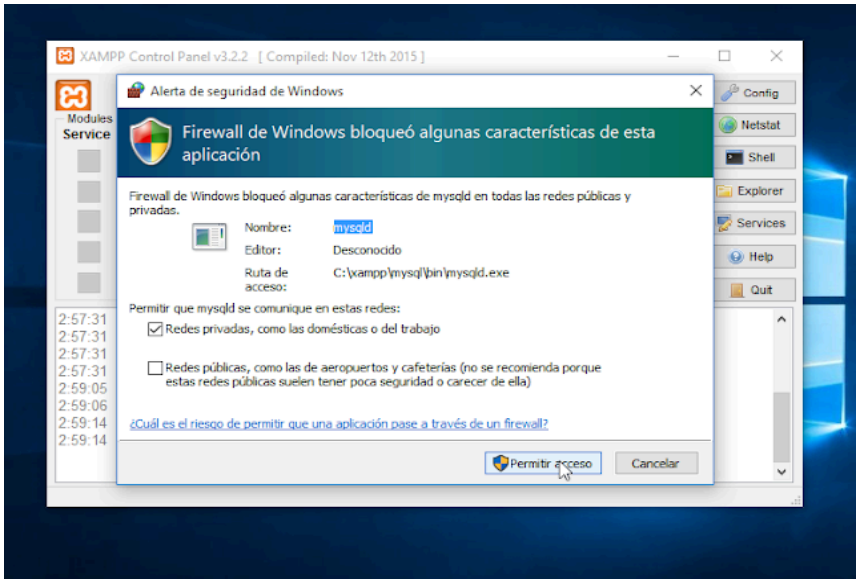


Figura 14. Permiso mysql

INSTALAR Y CONFIGURAR CODEIGNITER SOBRE XAMPP

INSTALACION CODEIGNITER

Para lograr la respectiva instalación y configuración de CodeIgniter sobre Xampp, deben seguir los puntos de a continuación:

1. Descargar CodeIgniter de este enlace: CodeIgniter. Versión 3.x

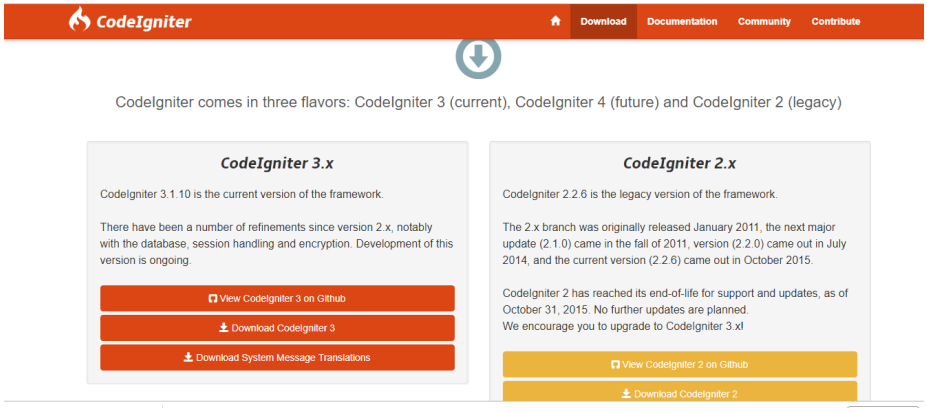


Figura 15. codeigniter

2. Ahora descomprimir el paquete en la carpeta htdocs, que se encuentra dentro de la carpeta xampp; al descomprimir se creará la carpeta CodeIgniter-3.3.10 con todos los archivos que trae este framework.

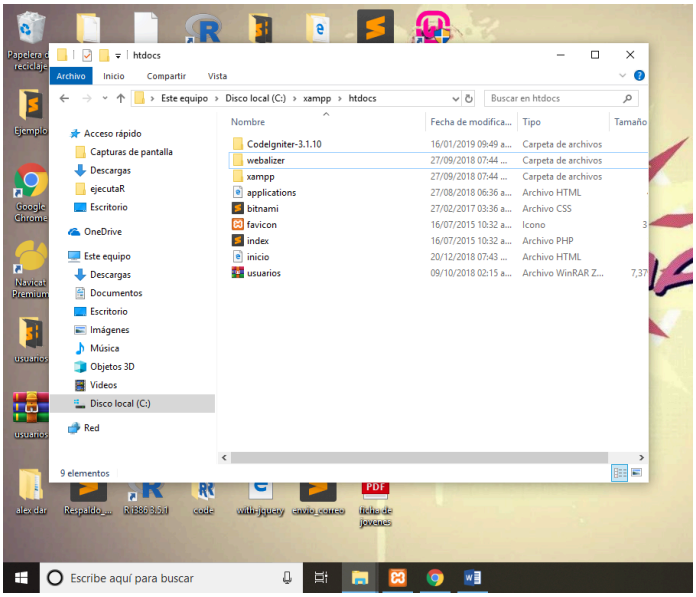


Figura 16. codeigniter extracción

Luego cambiamos el nombre de la carpeta CodeIgniter-3.3.10 por el nombre de nuestro proyecto "Residencia2018".

CONFIGURACION CODEIGNITER

Se ubican las carpetas de configuración

Para empezar, se accede a la carpeta de application que contiene los siguientes archivos.

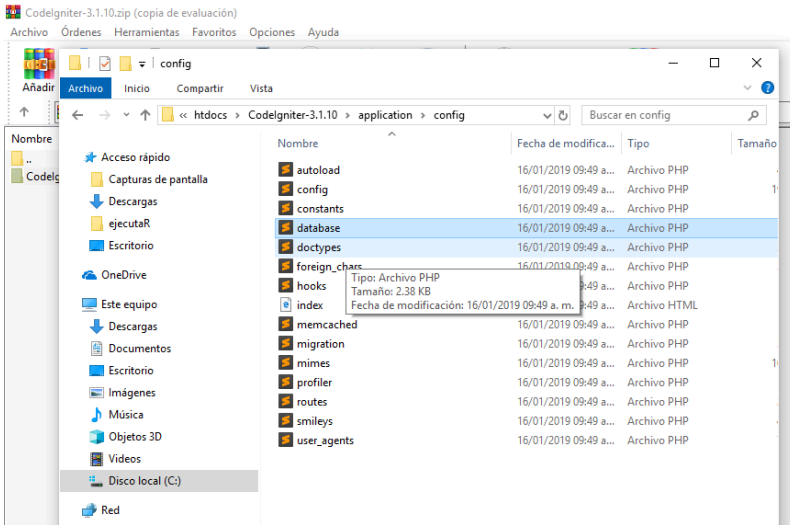


Figura 17. codeigniter config.

Archivo config.php

En el archivo config.php vamos a modificar la variable $config['base_url'], aquí indicamos la url del proyecto, la cual nos va a permitir acceder a este:

$config['base_url'] = 'http://localhost/Residencia2018v1

Archivo database.php

Ahora en el archivo database.php, si nuestro proyecto requiere de una base de datos, vamos a indicar el hostname, username, password y database a usar:

//El hostname por lo general es localhost.
$db['default']['hostname'] = 'localhost';
//El username es mi caso es root, si tienen otro usuario de base de datos, colocan ese nombre aquí.
$db['default']['username'] = 'root';
//En este caso al usuario no le eh colocado password por eso está vacío, caso contrario si su usuario tiene password lo colocan aquí.
$db['default']['password'] = '';

//Aquí colocamos el nombre de la base de datos, en mi caso es residencia2018v1.
$db['default']['database'] = ' residencia2018v1';

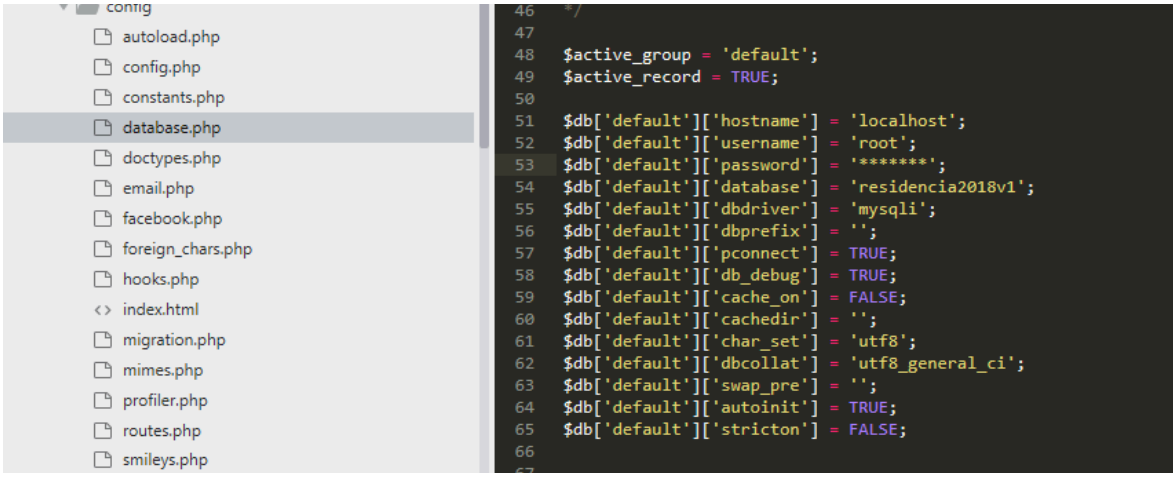


Figura 18.database

Finalmente probaremos si ya tenemos acceso a nuestro proyecto, para ello iniciamos nuestro xampp, vamos al navegador de nuestra preferencia y colocamos la url del proyecto. Si todo está bien se mostrará una ventana similar a lo que se aprecia en la imagen.

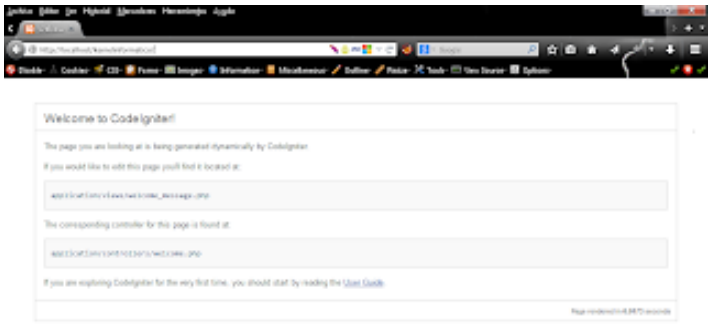


Figura 19. Prueba navegador

CONFIGURACIÓN PARA EL DISEÑO Y APLICACIÓN DE LA PLATAFORMA

Para el desarrollo del proyecto, debe tenerse en cuenta que CODEIGNITER es un FRAMEWORK basado en el Modelo-Vista-Controlador.
Se trabaja sobre una misma base de datos, como ya se vio en la parte de la configuración del Framework

DISEÑO Y ESTRUCURA DE LA BASE DE DATOS

La plataforma cuenta con Cuatro módulos.

- Login (Tanto para Admin como para Usuarios tipo Turistas)
- Productos Turísticos
- Búsqueda y Recomendación de Productos Turísticos
- Dashboard

Modulo del Sistema de Login-Diccionario de Datos

Usuario Administrador

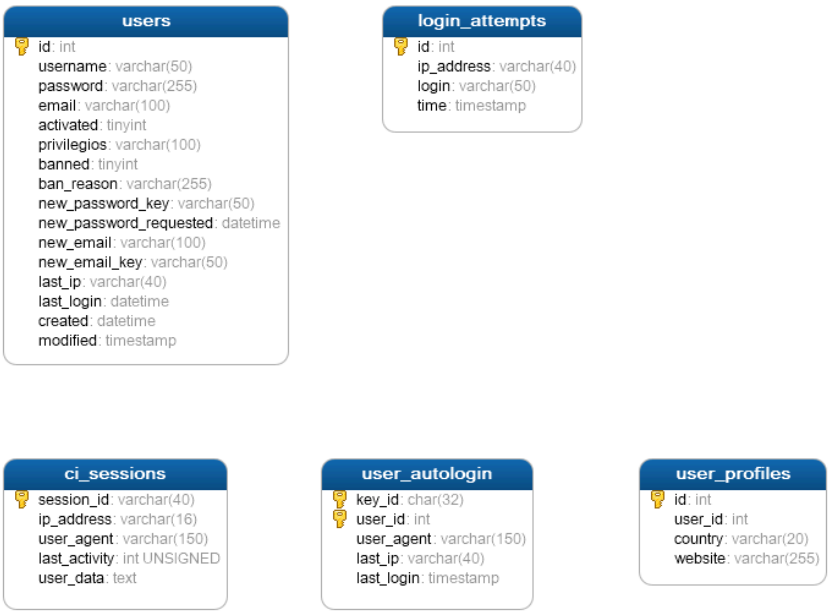


Figura 20.Relacional users

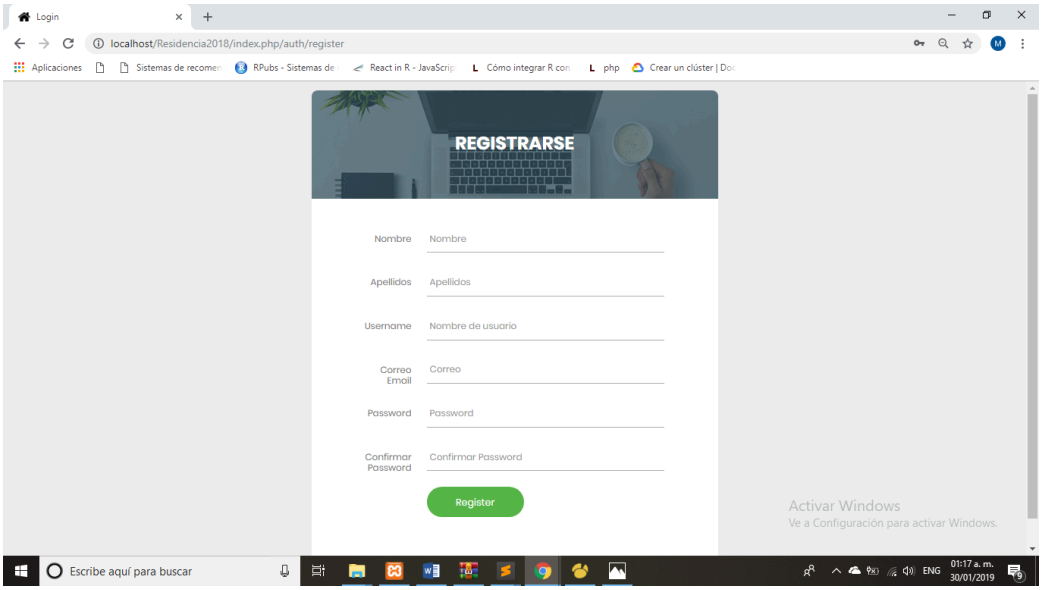


Figura 21. Registro

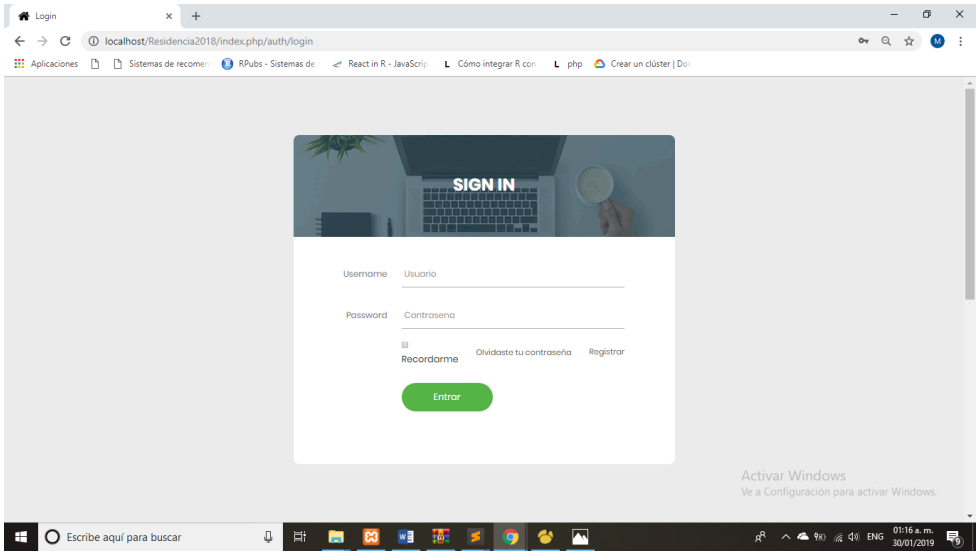


Figura 22. Inicio sesion

Ci_sessions	**TABLA DE CONTROL DE INICIO DE SESIONES**				
	Columna	Tipo	Nulo	predeterminado	descripción
	Session_id	Varchar(40)	no	0	Clave principal
	Ip_address	Varchar(40)	No	0	Dirección
	User_agent	Varchar(40)	No		Cabecera para identificar protocolo de red.
	Last_activity	Int(10)	No	0	Ultimo identificador
	User_data	Text	No		Datos de sesión
indices					

	primary	BTREE	No		Clave primaria
Login_attempts			No		
	Id	Int(11)	No		Clave primaria
	Ip_address	Varchar(40)	No		Dirección ip de usuario
	Login	Varchar(50)	No		Nombre de identificador del usuario para inicio de sesion.
	Time	Timestamp	No		Hora de inicio de sesión

Diccionario Datos. 1

USERS	TABLA DE REGISTRO DE NUEVOS USUARIOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id	Int(11)	no		Clave principal
	Username	Varchar(50)	No		Nombre del usuario
	Password	Varchar(255)	No		Password usuario
	Email	Varchar(100)	No		Email del usuario
	Activated	tinyint(1)	No	1	Verificar que el usuario este en modo activo.
	Privilegios	Varchar(100)	No		Asignación de roles de los usuarios
	Banned	tinyint(1)	No	0	Detecta si el usuario está bloqueado.
	Ban_reason	Varchar(255)	Si	Null	Razón por la cual se dio de baja un usuario

Diccionario Datos. 2

USERS	TABLA DE REGISTRO DE NUEVOS USUARIOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id	Int(11)	No		Clave principal
	Username	Varchar(50)	No		Nombre del usuario
	Password	Varchar(255)	No		Password usuario
	Email	Varchar(100)	No		Email del usuario
	Activated	tinyint(1)	No	1	Verificar que el usuario este en modo activo.
	Privilegios	Varchar(100)	No		Asignación de roles de los usuarios
	Banned	tinyint(1)	No	0	Detecta si el usuario está bloqueado.
	Ban_reason	Varchar(255)	Si	Null	Razón por la cual se dio de baja un usuario

Diccionario Datos. 3

USERS_AUTOLOGIN	TABLA DE REGISTRO DE NUEVOS USUARIOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Key_Id	Int(11)	No		Clave principal
	User_Id	Varchar(50)	No		Clave Foránea
	User_agent	Varchar(255)	No		Cabecera para identificar protocolo de red.
	Last_ip	Varchar(100)	No		Ultimo identificador
	Last_login	tinyint(1)	No	1	Ultimo inicio de sesión

Diccionario Datos. 4

USERS_PROFILES	TABLA PERFILES DE USUARIOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id	Int(11)	No		Clave principal
	User_Id	Varchar(50)	No		Clave Foránea Usuario
	Country	Varchar(255)	Null		País del usuario
	Website	Varchar(100)	Null		Sitio web

Diccionario Datos. 5

Usuario Final (Turista)

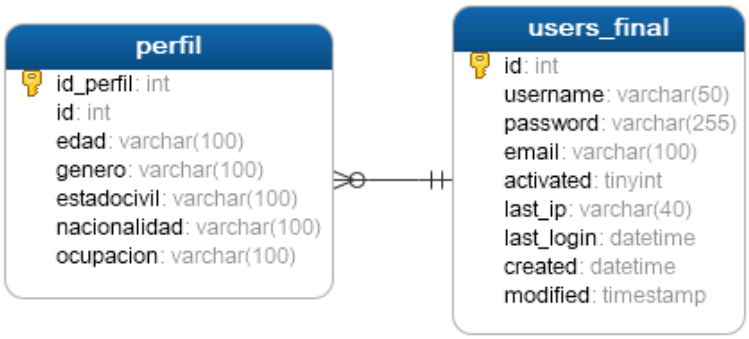


Figura 23. Perfil Turista

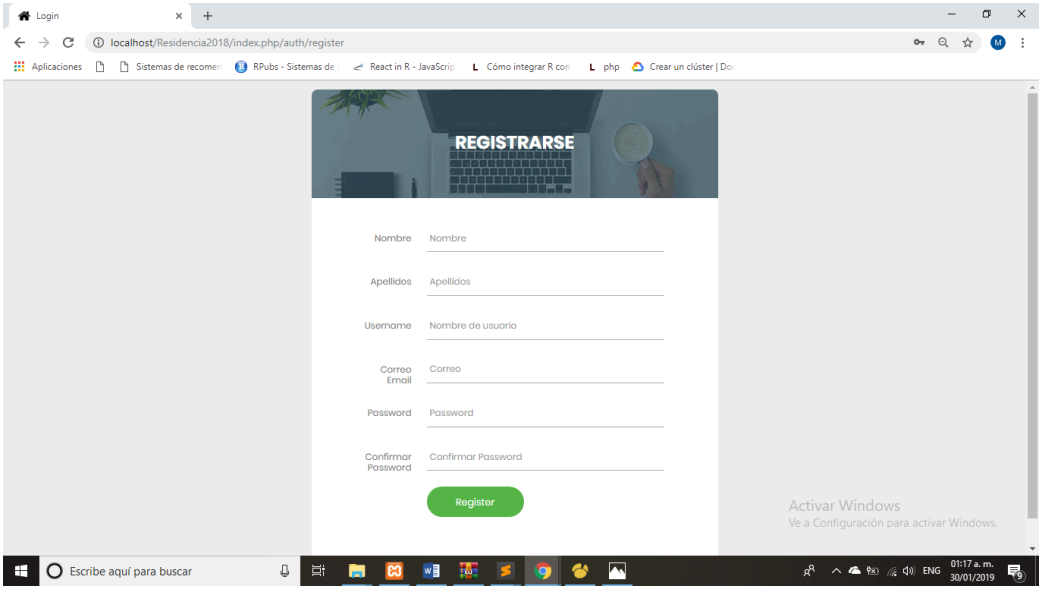


Figura 24. Registro Turista

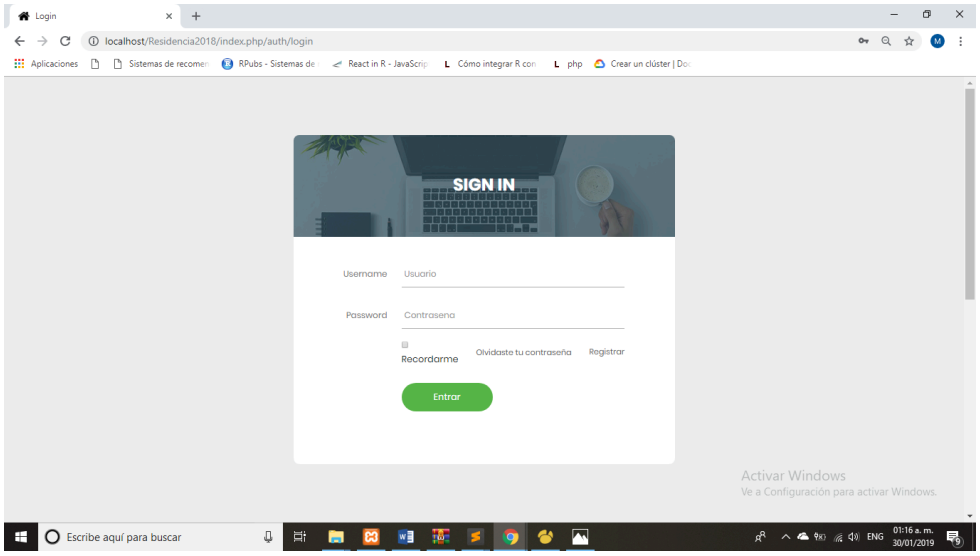


Figura 25. Inicio sesión Turista

USERS_FINAL	TABLA PERFILES DE USUARIOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id	Int(11)	No		Clave principal
	Username	Varchar(50)	No		Clave Foránea Usuario
	password	Varchar(255)	No		Contraseña de usuario
	email	Varchar(100)	No		Correo electrónico del usuario

Diccionario Datos. 6

PERFIL	TABLA DE LOS PERFILES DE LOS USUARIOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_perfil	Int(11)	No		Clave principal
	Id	Int(11)	No		Clave del usuario final
	Edad	Varchar(50)	Si		Edad del usuario final
	genero	Varchar(255)	Si		Genero del usuario final
	estadocivil	Varchar(100)	Si		Estado civil actual del usuario final
	nacionalidad	Varchar(100)	Si		Nacionalidad del usuario final
	ocupación	Varchar(100)	Si		Ocupación del usuario final

Diccionario Datos. 7

Módulo de Productos Turísticos-Diccionario de Datos

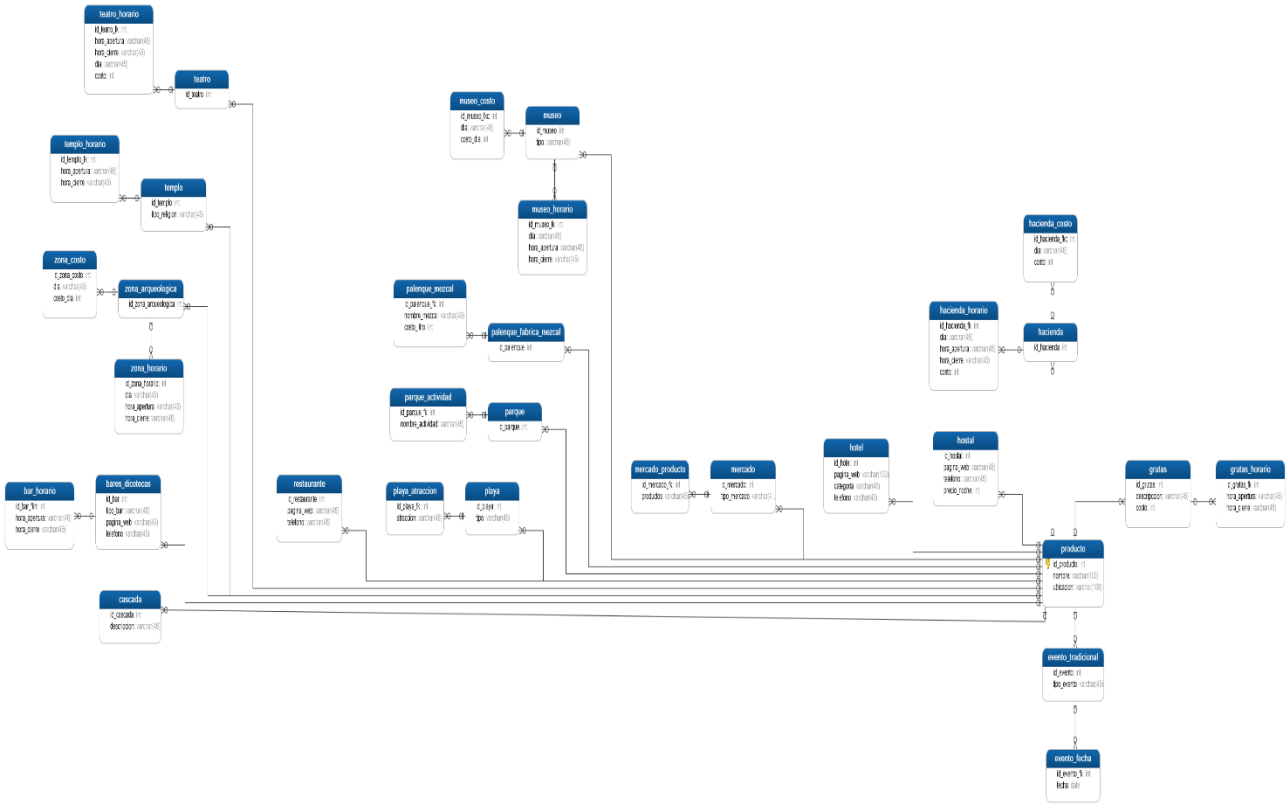


Figura 26. E-R

BARES_DISCOTECA	TABLA DE LOS BARES Y DISCOTECAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_bar	Int(11)	No		Identificador del bar y ó discoteca
	Tipo_bar	Varchar(50)	No		Tipo de bar
	Pagina_web	Varchar(50)	Si		Pagina web del bar
	telefono	Varchar(255)	Si		Teléfono del bar

Diccionario Datos. 8

BAR_HORARIO	TABLA DE LOS HORARIOS DEL BAR				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_bar_fk	Int(11)	No		Clave foránea del bar_discoteca

	Hora_apertura	Varchar(50)	No		Hora en la que abre el bar
	Hora_cierre	Varchar(50)	Si		Hora en la que cierra el bar

Diccionario Datos. 9

CASCADA	TABLA DE LAS CASCADAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_cascada	Int(11)	No		Identificador del bar y ó discoteca
	descripcion	Varchar(50)	No		Descripción de la cascada

Diccionario Datos. 10

Evento_fecha	Tabla de las fechas de los eventos				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_evento_fk	Int(11)	No		Clave foránea del evento
	fecha	Date	No		Fecha en la que se realizara el evento

Diccionario Datos. 11

EVENTO_TRADICIONAL	TABLA DE LOS BARES Y DISCOTECAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_evento	Int(11)	No		Identificador del evento
	Tipo_evento	Varchar(50)	No		Tipo de evento

Diccionario Datos. 12

GRUTAS	TABLA DE GRUTAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_grutas	Int(11)	No		Identificador del gruta
	descripcion	Varchar(50)	No		Descripción de la gruta
	costo	Int(11)			Costo de acceso

Diccionario Datos. 13

GRUTAS_HORARIO	TABLA HORARIOS DE GRUTAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_grutas_fk	Int(11)	No		Identificador del evento
	Hora_apertura	Varchar(50)	No		Hora en la que abren las grutas
	Hora_cierre	Varchar(50)	No		Hora en la que cierran las grutas

Diccionario Datos. 14

HACIENDA	TABLA HACIENDA				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_hacienda	Int(11)	No		Identificador de la hacienda

Diccionario Datos. 15

HACIENDA_COSTO	TABLA COSTO ACCESO HACIENDA				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_hacienda_fkc	Int(11)	No		Clave foránea hacienda
	Dia	Varchar(50)	No		Días de la semana que abren
	costo	Int(11)	No		Costo de acceso
	Hora_apertura	Varchar(50)	No		Hora de apertura de la hacienda
	Hora_cerre	Varchar(50)	no		Hora de cierre de la hacienda

Diccionario Datos. 16

HOSTAL	TABLA DE HOSTALES				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_hostal	Int(11)	No		Clave principal de hostales
	Pagina_web	Varchar(50)	No		Pagina web del hostal
	telefono	Varchar(50)	No		Teléfono del hostal
	Precio_noche	Varchar(50)			Precio por noche del hostal

Diccionario Datos. 17

HOTEL	TABLA DE HOTELES				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_hotel	Int(11)	No		Clave principal de hotel
	Pagina_web	Varchar(50)	No		Pagina web del hotel
	telefono	Varchar(50)	No		Teléfono del hotel
	categoriia	Varchar(50)			Cuantas estrellas tiene

Diccionario Datos. 18

MERCADO	TABLA DE MERCADOS

	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_mercado	Int(11)	No		Clave principal del mercado
	Tipo_mercado	Varchar(50)	No		Tipo de mercado que es

Diccionario Datos. 19

MERCADO_PRODUCTOS	TABLA DELOS PRODUCTOS DE LOS MERCADOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_mercado_fk	Int(11)	No		Clave foranea mercado
	productos	Varchar(50)	No		Tipo de productos que ofrece

Diccionario Datos. 20

MUSEO	TABLA DE LOS MUSEOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_museo	Int(11)	No		Clave principal del museo
	Tipo	Varchar(50)	No		Tipo de museo

Diccionario Datos. 21

MUSEO_HORARIO	TABLA DE HORARIOS Y COSTOS DE LOS MUSEOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_museo_fk	Int(11)	No		Clave foránea museo
	dia	Varchar(50)	No		Dias que abre el museo
	Hora_apertura	Varchar(50)	No		Hora a la que abre el museo
	Hora_cierre	Varchar(50)	no		Hora a la que cierra el museo

Diccionario Datos. 22

PALENQUE_FABRICA_MEZCAL	TABLA DE PALENQUES Y FABRICAS DE MEZCAL				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_palenque	Int(11)	No		Clave foránea museo

Diccionario Datos. 23

PALENQUE_MEZCAL	TABLA DE COSTOS POR LITRO DE MEZCAL

	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_palenque_fk	Int(11)	No		Clave foránea palenque
	nombre	Varchar(50)	No		Nombre mezcal
	costo litro	Varchar(50)	No		Precio por litro

Diccionario Datos. 24

PARQUE	TABLA DE LOS PARQUES				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_parque	Int(11)	No		Clave principal parque

Diccionario Datos. 25

PARQUE ACTIVIDAD	TABLA DE ACTIVIDADES DE LOS PARQUES				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_parque_fk	Int(11)	No		Clave foránea parque
	Nombre_actividad	Varchar(50)	No		Actividades que realizan en el parque

Diccionario Datos. 26

PLAYA	TABLA PLAYAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_playa	Int(11)	No		Clave principal playa
	tipo	Varchar(50)	No		Tipo de playa que es

Diccionario Datos. 27

PLAYA_ATRACCION	TABLA PLAYAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_playa_fk	Int(11)	No		Clave foránea playa
	atraccion	Varchar(50)	No		Actividades que hay en la playa

Diccionario Datos. 28

PRODUCTO	TABLA DE LOS PRODUCTOS TURISTICOS				
	Columna	Tipo	Nulo	predeterminado	Descripción

	Id_producto	Int(11)	No		Clave principal producto
	nombre	Varchar(50)	No		Nombre del producto
	ubicacion	Varchar(250)	No		Ubicación de los productos turisticos

Diccionario Datos. 29

RESTAURANTE	TABLA DE LOS RESTAURANTES				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_playa	Int(11)	No		Clave principal restaurante
	Pagina_web	Varchar(50)	No		Pagina del restaurante
	telefono	Varchar(50)	No		Teléfono del restaurante

Diccionario Datos. 30

TEATRO_HORARIO	TABLA DE HORARIOS Y COSTOS DE LOS TEATROS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_teatro_fk	Int(11)	No		Clave foránea teatro
	dia	Varchar(50)	No		Dias que abre el teatro
	Hora_apertura	Varchar(50)	No		Hora a la que abre el teatro
	Hora_cierre	Varchar(50)	No		Hora a la que cierra el teatro
	costo	Int(11)	No		Costo de acceso al teatro

Diccionario Datos. 31

TEMPLO	TABLA DE LOS TEMPLOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_templo	Int(11)	No		Clave principal tempo
	Tipo_religion	Varchar(50)	No		Tipo de religión que es el templo o iglesia

Diccionario Datos. 32

TEMPLO_HORARIO	TABLA DE HORARIOS DE LOS TEMPLOS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_templo_fk	Int(11)	No		Clave foránea templo

	Hora_apertura	Varchar(50)	No		Hora a la que abre el templo
	Hora_cierre	Varchar(50)	no		Hora a la que cierra el templo

Diccionario Datos. 33

ZONA	TABLA DE LAS ZONAS ARQUOLOGICAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_zona_arqueologiac	Int(11)	No		Clave principal zona arqueologica

Diccionario Datos. 34

ZONA_HORARIO	TABLA DE HORARIOS DE LAS ZONAS ARQUOLOGICAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_zona_fk	Int(11)	No		Clave foránea zona arquologica
	dia	Varchar(50)	No		Dias que abre el zona arquologica
	Hora_apertura	Varchar(50)	No		Hora a la que abre el zona arquologica
	Hora_cierre	Varchar(50)	no		Hora a la que cierra el zona arqueologica

Diccionario Datos. 35

ZONA_COSTO	TABLA DE HORARIOS DE LAS ZONAS ARQUOLOGICAS				
	Columna	Tipo	Nulo	predeterminado	Descripción
	Id_zona_fk	Int(11)	No		Clave foránea zona arquologica
	dia	Int(11)	No		Dias que abre el zona arqueologica
	Costo_dia	Varchar(50)	No		Costo de acceso a la zona arqueologica

Diccionario Datos. 36

Descripción de las clases del módulo de productos turísticos

Controladores

Controlador bar

Tabla 1 Controlador bar

Nombre archivo	Funciones	Descripción	Ubicación
bar.php	function__construct()	Es el método constructor	controllers/bar

	index()	Si esta autenticado cargara la vista de agregar_bar	
	crearBar()	Obtiene los parámetros de la vista agregar_bar y validarlos para agregarlos	
	listarbar()	Crear los parámetros, y links para la paginación de los productos,	
	show_bar_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_bar_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	update_bar()	Obtiene los parámetros de la vista actualizar_bar para envíralos al modelo.	
	delete_bar_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador cascada

Tabla 2 Controlador cascada

Nombre archivo	Funciones	Descripción	Ubicación
cascada.php	function__construct( )	Es el método constructor	controllers/cascada
	index()	Si esta autenticado cargara la vista de agregar_cascada	
	crearCascada()	Obtiene los parámetros de la vista agregar_cascada y validarlos para agregarlos	
	listarcascada()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_cascada_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_cascada_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_cascada()	Obtiene los parámetros de la vista actualizar_cascada para envíralos al modelo.	

	delete_cascada_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador evento tradicional

Tabla 3 Controlador Evento Tradicional

Nombre archivo	Funciones	Descripción	Ubicación
evento.php	function__construct( )	Es el método constructor	controllers/evento
	index()	Si esta autenticado cargara la vista de agregar_evento	
	crearEvento()	Obtiene los parámetros de la vista agregar_evento y validarlos para agregarlos	
	listarevento()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_evento_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_evento_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_evento()	Obtiene los parámetros de la vista actualizar_evento para envíralos al modelo.	
	delete_evento_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador gruta

Tabla 4 Controlador gruta

Nombre archivo	Funciones	Descripción	Ubicación
gruta.php	function__construct( )	Es el método constructor	controllers/gruta
	index()	Si esta autenticado cargara la vista de agregar_gruta	
	crearGruta()	Obtiene los parámetros de la vista agregar_gruta y validarlos para agregarlos	
	listargruta()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_gruta_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_gruta_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_gruta()	Obtiene los parámetros de la vista actualizar_gruta para envíralos al modelo.	
	delete_gruta_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador hacienda

Tabla 5 Controlador hacienda

Nombre archivo	Funciones	Descripción	Ubicación
hacienda.php	function__construct()	Es el método constructor	controllers/hacienda
	index()	Si esta autenticado cargara la vista de agregar_hacienda	
	crearHacienda()	Obtiene los parámetros de la vista agregar_hacienda y validarlos para agregarlos	
	listarhacienda()	Crear los parámetros, y links para la paginación de los productos cascadas,	

	show_hacienda_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_hacienda_id_d( )	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_hacienda()	Obtiene los parámetros de la vista actualizar_hacienda para envíralos al modelo.	
	delete_hacienda_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador hostal

Tabla 6 Controlador hostal

Nombre archivo	Funciones	Descripción	Ubicación
hostal.php	function__construct( )	Es el método constructor	controllers/hostal
	index()	Si esta autenticado cargara la vista de agregar_hostal	
	crearhostal()	Obtiene los parámetros de la vista agregar_hostal y validarlos para agregarlos	
	listarhostal()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_hostal_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_hostal_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	update_hostal()	Obtiene los parámetros de la vista actualizar_hostal para envíralos al modelo.	
	delete_hostal_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	

	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador hotel

Tabla 7 Controlador hotel

Nombre archivo	Funciones	Descripción	Ubicación
hotel.php	function__construct( )	Es el método constructor	controllers/hotel
	index()	Si esta autenticado cargara la vista de agregar_hotel	
	crearCascadar()	Obtiene los parámetros de la vista agregar_hotel y validarlos para agregarlos	
	listarhotel()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_hotel_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_hotel_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_hotel()	Obtiene los parámetros de la vista actualizar_hotel para envíralos al modelo.	
	delete_hotel_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador mercado

Tabla 8 Controlador mercado

Nombre archivo	Funciones	Descripción	Ubicación
mercado.php	function__construct()	Es el método constructor	controllers/mercado
	index()	Si esta autenticado cargara la vista de agregar_mercado	
	crearMercado()	Obtiene los parámetros de la vista agregar_mercado y validarlos para agregarlos	

	listarmercado()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_cascada_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_mercado_id_d( )	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_mercado()	Obtiene los parámetros de la vista actualizar_mercado para envíralos al modelo.	
	delete_mercado_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador museo

Tabla 9 Controlador museo

Nombre archivo	Funciones	Descripción	Ubicación
museo.php	function__construct( )	Es el método constructor	controllers/museo
	index()	Si esta autenticado cargara la vista de agregar_museo	
	crearMuseo()	Obtiene los parámetros de la vista agregar_museo y validarlos para agregarlos	
	listarmuseo()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_museo_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_museo_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	

	Update_museo()	Obtiene los parámetros de la vista actualizar_museo para envíralos al modelo.	
	delete_museo_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador palenque

Tabla 10 Controlador palenque

Nombre archivo	Funciones	Descripción	Ubicación
palenque.php	function__construct()	Es el método constructor	controllers/palenque
	index()	Si esta autenticado cargara la vista de agregar_palenque	
	crearPalenque()	Obtiene los parámetros de la vista agregar_palenque y validarlos para agregarlos	
	listarpalenque()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_palenque_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_palenque_id_d( )	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_palenque()	Obtiene los parámetros de la vista actualizar_palenque para envíralos al modelo.	
	delete_palenque_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador parque

Tabla 11 Controlador parque

Nombre archivo	Funciones	Descripción	Ubicación
parque.php	function__construct( )	Es el método constructor	controllers/parque
	index()	Si esta autenticado cargara la vista de agregar_parque	
	crearParque()	Obtiene los parámetros de la vista agregar_parque y validarlos para agregarlos	
	listarparque()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_parque_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_parque_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_parque()	Obtiene los parámetros de la vista actualizar_parque para envíralos al modelo.	
	delete_parque_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador playa

Ilustración 57 Controlador playa

Nombre archivo	Funciones	Descripción	Ubicación
playa.php	function__construct( )	Es el método constructor	controllers/playa
	index()	Si esta autenticado cargara la vista de agregar_playa	

	crearPlaya()	Obtiene los parámetros de la vista agregar_playa y validarlos para agregarlos	
	listarplaya()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_playa_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_playa_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_playa()	Obtiene los parámetros de la vista actualizar_playa para envíralos al modelo.	
	delete_playa_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador restaurante

Tabla 12 Controlador restaurante

Nombre archivo	Funciones	Descripción	Ubicación
restaurante.php	function__construct()	Es el método constructor	controllers/restaurante
	index()	Si esta autenticado cargara la vista de agregar_restaurante	
	crearRestaurante()	Obtiene los parámetros de la vista agregar_restaurante y validarlos para agregarlos	
	listarrestaurante()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_restaurante_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_restaurante_id_d()	Recibe como parámetro el id del producto para poder mostrar su	

		información y llamar a la función delete	
	Update_restaurante()	Obtiene los parámetros de la vista actualizar_restaurante para envíralos al modelo.	
	delete_restaurante_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador teatro

Tabla 13 Controlador teatro

Nombre archivo	Funciones	Descripción	Ubicación
teatro.php	function__construct( )	Es el método constructor	controllers/teatro
	index()	Si esta autenticado cargara la vista de agregar_teatro	
	crearTeatro()	Obtiene los parámetros de la vista agregar_teatro y validarlos para agregarlos	
	listarteatro()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_teatro_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_teatro_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_teatro()	Obtiene los parámetros de la vista actualizar_teatro para envíralos al modelo.	
	delete_teatro_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador templo

Tabla 14 Controlador templo

Nombre archivo	Funciones	Descripción	Ubicación
templo.php	function__construct( )	Es el método constructor	controllers/templo
	index()	Si esta autenticado cargara la vista de agregar_templo	
	crearTemplo()	Obtiene los parámetros de la vista agregar_templo y validarlos para agregarlos	
	listartemplo()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_templo_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_templo_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_templo()	Obtiene los parámetros de la vista actualizar_templo para envíralos al modelo.	
	delete_templo_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Controlador zona arqueológica

Tabla 15 Controlador zona arqueológica

Nombre archivo	Funciones	Descripción	Ubicación
zona.php	function__construct( )	Es el método constructor	controllers/zona
	index()	Si esta autenticado cargara la vista de agregar_zona	

	crearZona()	Obtiene los parámetros de la vista agregar_zona y validarlos para agregarlos	
	listarzona()	Crear los parámetros, y links para la paginación de los productos cascadas,	
	show_zona_id()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función actualizar	
	show_zona_id_d()	Recibe como parámetro el id del producto para poder mostrar su información y llamar a la función delete	
	Update_zona()	Obtiene los parámetros de la vista actualizar_zona para envíralos al modelo.	
	delete_zona_id()	Recibe como parámetro el id y llamar a la función del modelo para eliminarlo	
	custom()	Carga el modelo, inicia los parámetros, obtiene los registros, hace la configuración de la paginación	

Modelos

Modelo bar

Tabla 16 Modelo bar

Nombre archivo	Funciones	Descripción	Ubicación
barmodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosBar($limite = 0)	Obtiene los datos de todos bares	
	show_bars()	Obtiene lista de los nombres de los bares	
	show_bar_id($data)	Obtiene un bar a partir del id que se pasa	
	update_bar($id,$data)	Actualiza la tabla bares y discotecas pasándole el id y los parámetros	
	show_bar_id_d($data)	Obtiene una lista de los nombre de los bares que puede eliminar	
	delete_bar_id($id)	Dado el id elimina de la tabla el registro.	

Modelo cascada

Tabla 17 Modelo cascada

Nombre archivo	Funciones	Descripción	Ubicación
cascadamodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosCascada($limite = 0)	Obtiene los datos de todas las cascadas	
	show_cascadas()	Obtiene lista de los nombres de las cascadas	
	show_cascada_id($data)	Obtiene un registro de una cascada a partir del id que se está buscando	
	update_cascada($id,$data)	Actualiza la tabla cascadas y pasándole el id de la tabla a actualizar y los parámetros	
	show_cascada_id_d($data)	Obtiene una lista de los nombre de las cascada para ver pasa su eliminación	
	delete_cascada_id($id)	Dado el id elimina de la tabla cascadas un registro	

Modelo evento tradicional

Tabla 18 Modelo evento tradicional

Nombre archivo	Funciones	Descripción	Ubicación
eventomodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosBar($limite = 0)	Obtiene los datos de todos eventos	
	show_eventoss()	Obtiene lista de los nombres de los eventos	
	show_evento_id($data)	Obtiene un evento a partir del id que se está buscando	
	update_evento($id,$data)	Actualiza la tabla eventos tradicionales pasándole el id de la tabla a actualizar y los parámetros	
	show_evento_id_d($data)	Obtiene una lista de los nombres de los eventos para ver pasa su eliminación	
	delete_evento_id($id)	Dado el id elimina de la tabla el registro de evento tradicional	

Modelo grutas

Tabla 19 Modelo grutas

Nombre archivo	Funciones	Descripción	Ubicación
grutamodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosGruta($limite = 0)	Obtiene los datos de todas las grutas	
	show_grutas()	Obtiene lista de los nombres de las grutas	
	show_gruta_id($data)	Obtiene una gruta a partir del id que se está buscando	
	update_gruta($id,$data)	Actualiza la tabla grutas pasándole el id del registro y los parámetros	
	show_gruta_id_d($data)	Obtiene una lista de los nombres de las grutas que va a eliminar	
	delete_gruta_id($id)	Dado el id elimina de la tabla grutas un registro	

Modelo hacienda

Ilustración 58 Modelo hacienda

Nombre archivo	Funciones	Descripción	Ubicación
haciendamodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosHacienda($limite = 0)	Obtiene los datos de todas las haciendas	
	show_haciendas()	Obtiene lista de los nombres de las haciendas	
	show_hacienda_id($data)	Obtiene una hacienda a partir del id que se está buscando	
	update_hacienda($id,$data)	Actualiza la hacienda pasándole el id de la tabla a actualizar y los parámetros	
	show_hacienda_id_d($data)	Obtiene una lista de los nombres de las haciendas que pueden ser eliminadas	
	delete_hacienda_id($id)	Dado el id elimina de la tabla hacienda un registro	

Modelo hostal

Tabla 20 Modelo hostal

Nombre archivo	Funciones	Descripción	Ubicación
hostalmodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosHostal($limite = 0)	Obtiene los datos de todos los hostales	
	show_hostales()	Obtiene lista de los nombres de los hostales	
	show_hostales_id($data)	Obtiene un hostal a partir del id que se está buscando	
	update_hostal($id,$data)	Actualiza la hostal pasándole el id de la tabla y los parámetros	
	show_hostal_id_d($data)	Obtiene una lista de los nombre de los hostales que pueden ser eliminados	
	delete_hostal_id($id)	Dado el id elimina de la tabla el hostales un registro	

Modelo hotel

Tabla 21 Modelo hotel

Nombre archivo	Funciones	Descripción	Ubicación
hotelmodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosHotel($limite = 0)	Obtiene los datos de todos los hoteles	
	show_hoteles()	Obtiene lista de los nombres de los hoteles	
	show_hoteles_id($data)	Obtiene un hotel a partir del id que se está buscando	
	update_hotel($id,$data)	Actualiza el hotel pasándole el id de la tabla y los parámetros	
	show_hotel_id_d($data)	Obtiene una lista de los nombres de los hoteles que pueden ser eliminados	
	delete_hoetel_id($id)	Dado el id elimina de la tabla el hoteles un registro	

Modelo mercado

Tabla 22 Modelo mercado

Nombre archivo	Funciones	Descripción	Ubicación
mercadomodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosMercadol($limite = 0)	Obtiene los datos de todos los mercados	
	show_mercados()	Obtiene lista de los nombres de los mercados	
	show_mercados_id($data)	Obtiene un mercado a partir del id que se está buscando	
	update_mercado($id,$data)	Actualiza la tabla mercado pasándole el id y los parámetros	
	show_mercado_id_d($data)	Obtiene una lista de los nombre de los mercados que pueden ser eliminados	
	delete_mercado_id($id)	Dado el id elimina de la tabla el mercado un registro	

Modelo museos

Tabla 23 Modelo museos

Nombre archivo	Funciones	Descripción	Ubicación
museomodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosMuseo($limite = 0)	Obtiene los datos de todos museos	
	show_museos()	Obtiene lista de los nombres de los museos	
	show_museos_id($data)	Obtiene un museo a partir del id que se está buscando	
	update_museo($id,$data)	Actualiza la tabla museo pasándole el id y los parámetros	
	show_museo_id_d($data)	Obtiene una lista de los nombres de los museos que pueden ser eliminados	
	delete_museo_id($id)	Dado el id elimina de la tabla el museo un registro	

Modelo palenque

Tabla 24 Modelo Palenque

Nombre archivo	Funciones	Descripción	Ubicación

palenquemodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosPalenque($limite = 0)	Obtiene los datos de todos palenques	
	show_palenques()	Obtiene lista de los nombres de los palenques	
	show_palenques_id($data)	Obtiene un palenque a partir del id que se está buscando	
	update_palenque($id,$data)	Actualiza la tabla palenque pasándole el id y los parámetros	
	show_palenque_id_d($data)	Obtiene una lista de los nombre de los palenques que pueden ser eliminados	
	delete_palenque_id($id)	Dado el id elimina de la tabla el palenque un registro	

Modelo parques

Tabla 25 Modelo parques

Nombre archivo	Funciones	Descripción	Ubicación
parquemodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosParque($limite = 0)	Obtiene los datos de todos parques	
	show_parques ()	Obtiene lista de los parques de los hostales	
	show_parques_id($data)	Obtiene un parque a partir del id que se está buscando	
	update_parque($id,$data)	Actualiza la tabla parque pasándole el id y los parámetros	
	show_parque_id_d($data)	Obtiene una lista de los nombre de los parques que pueden ser eliminados	
	delete_parque_id($id)	Dado el id elimina de la tabla el parques un registro	

Modelo playa

Ilustración 59 Modelo playa

Nombre archivo	Funciones	Descripción	Ubicación
playamodel.php	function __construct()	Constructor	/models/

	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosPlaya($limite = 0)	Obtiene los datos de todos playas	
	show_playas()	Obtiene lista de los nombres de los playas	
	show_playas_id($data)	Obtiene un playa a partir del id que se está buscando	
	update_playa($id,$data)	Actualiza la tabla playa pasándole el id y los parámetros	
	show_palya_id_d($data)	Obtiene una lista de los nombres de las playas que pueden ser eliminados	
	delete_playa_id($id)	Dado el id elimina de la tabla el playas un registro	

Modelo restaurantes

Tabla 26 Modelo restaurantes

Nombre archivo	Funciones	Descripción	Ubicación
restaurantemodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosRsturante($limite = 0)	Obtiene los datos de todos restaurantes	
	show_restaurantes()	Obtiene lista de los nombres de los restaurantes	
	show_restaurantes_id($data)	Obtiene un restaurante a partir del id que se está buscando	
	update_restaurante($id,$data)	Actualiza la tabla restaurante pasándole el id y los parámetros	
	show_restaurante_id_d($data)	Obtiene una lista de los nombre de los restaurantes que pueden ser eliminados	
	delete_restaurante_id($id)	Dado el id elimina de la tabla restaurantes un registro	

Modelo teatro

Tabla 27 Modelo teatro

Nombre archivo	Funciones	Descripción	Ubicación

teatromodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosTeatro($limite = 0)	Obtiene los datos de todos teatros	
	show_teatros()	Obtiene lista de los nombres de los teatros	
	show_teatros_id($data)	Obtiene un teatro a partir del id que se está buscando	
	update_teatro($id,$data)	Actualiza la tabla teatro pasándole el id y los parámetros	
	show_teatro_id_d($data)	Obtiene una lista de los nombre de los teatros que pueden ser eliminados	
	delete_teatro_id($id)	Dado el id elimina de la tabla el teatro un registro	

Modelo templo

Tabla 28 Modelo templo

Nombre archivo	Funciones	Descripción	Ubicación
templomodel.php	function __construct()	Constructor	/models/
	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosTemplo($limite = 0)	Obtiene los datos de todos templos	
	show_templos()	Obtiene lista de los nombres de los templos	
	show_templos_id($data)	Obtiene un templo a partir del id que se está buscando	
	update_templo($id,$data)	Actualiza la tabla templo pasándole el id y los parámetros	
	show_templo_id_d($data)	Obtiene una lista de los nombres de los templos que pueden ser eliminados	
	delete_templo_id($id)	Dado el id elimina de la tabla el templos un registro	

Modelo zona arqueológica

Tabla 29 Modelo zona arqueológica

Nombre archivo	Funciones	Descripción	Ubicación
zonamodel.php	function __construct()	Constructor	/models/

	get_current_page_records($limit, $start)	Obtener registros de página actual	
	get_total()	Obtiene el número total de registros	
	verTodosZona($limite = 0)	Obtiene los datos de todas las zonas arqueológicas	
	show_zonas()	Obtiene lista de los nombres de las zonas	
	show_zonas_id($data)	Obtiene una zona a partir del id que se está buscando	
	update_zona($id,$data)	Actualiza la tabla zona arqueológicas pasándole el id y los parámetros	
	show_zona_id_d($data)	Obtiene una lista de los nombres de las zonas arqueológicas que pueden ser eliminados	
	delete_zona_id($id)	Dado el id elimina de la tabla zonas arqueológicas un registro	

Vistas

Tabla 30 Vistas

Nombre	Descripción	Ubicación
Agregar_bar.php	Formulario para un nuevo ingreso de un producto	/views/bar
Editar_bar.php	Vista y Formulario para editar producto	
Eliminar_bar.php	Vista de producto a eliminar	
Ver_bar.php	Vista de los productos	
Agregar_cascada.php	Formulario para un nuevo ingreso de un producto	/views/cascada
Editar_cascada.php	Vista y Formulario para editar producto	
Eliminar_cascada.php	Vista de producto a eliminar	
Ver_cascada.php	Vista de los productos	
Agregar_evento.php	Formulario para un nuevo ingreso de un producto	/views/evento
Editar_ evento.php	Vista y Formulario para editar producto	
Eliminar_ evento.php	Vista de producto a eliminar	
Ver_ evento.php	Vista de los productos	
Agregar_gruta.php	Formulario para un nuevo ingreso de un producto	/views/gruta
Editar_gruta.php	Vista y Formulario para editar producto	
Eliminar_gruta.php	Vista de producto a eliminar	
Ver_gruta.php	Vista de los productos	
Agregar_hacienda.php	Formulario para un nuevo ingreso de un producto	/views/hacienda
Editar_ hacienda.php	Vista y Formulario para editar producto	
Eliminar_ hacienda.php	Vista de producto a eliminar	
Ver_ hacienda.php	Vista de los productos	
Agregar_hostal.php	Formulario para un nuevo ingreso de un producto	/views/hostales
Editar_ hostal.php	Vista y Formulario para editar producto	
Eliminar_ hostal.php	Vista de producto a eliminar	
Ver_ hostal.php	Vista de los productos	
Agregar_hotel.php	Formulario para un nuevo ingreso de un producto	/views/hoteles
Editar_hotel.php	Vista y Formulario para editar producto	
Eliminar_hotel.php	Vista de producto a eliminar	
Ver_hotel.php	Vista de los productos	

Agregar_mercado.php	Formulario para un nuevo ingreso de un producto	/views/mercado
Editar_mercado.php	Vista y Formulario para editar producto	
Eliminar_mercado.php	Vista de producto a eliminar	
Ver_mercado.php	Vista de los productos	
Agregar_museo.php	Formulario para un nuevo ingreso de un producto	/views/museo
Editar_museo.php	Vista y Formulario para editar producto	
Eliminar_museo.php	Vista de producto a eliminar	
Ver_museo.php	Vista de los productos	
Agregar_palenque.php	Formulario para un nuevo ingreso de un producto	/views/palenque
Editar_palenque.php	Vista y Formulario para editar producto	
Eliminar_palenque.php	Vista de producto a eliminar	
Ver_palenque.php	Vista de los productos	
Agregar_parque.php	Formulario para un nuevo ingreso de un producto	/views/parque
Editar_parque.php	Vista y Formulario para editar producto	
Eliminar_parque.php	Vista de producto a eliminar	
Ver_parque.php	Vista de los productos	
Agregar_playa.php	Formulario para un nuevo ingreso de un producto	/views/playa
Editar_playa.php	Vista y Formulario para editar producto	
Eliminar_playa.php	Vista de producto a eliminar	
Ver_playa.php	Vista de los productos	
Agregar_restaurante.php	Formulario para un nuevo ingreso de un producto	/views/restaurant e
Editar_restaurante.php	Vista y Formulario para editar producto	
Eliminar_restaurante.ph p	Vista de producto a eliminar	
Ver_restaurante.php	Vista de los productos	
Agregar_taetro.php	Formulario para un nuevo ingreso de un producto	/views/teatro
Editar_teatro.php	Vista y Formulario para editar producto	
Eliminar_teatro.php	Vista de producto a eliminar	
Ver_teatro.php	Vista de los productos	
Agregar_templo.php	Formulario para un nuevo ingreso de un producto	/views/templos
Editar_templo.php	Vista y Formulario para editar producto	
Eliminar_templo.php	Vista de producto a eliminar	
Ver_templo.php	Vista de los productos	
Agregar_zona.php	Formulario para un nuevo ingreso de un producto	/views/zona
Editar_zona.php	Vista y Formulario para editar producto	
Eliminar_zona.php	Vista de producto a eliminar	
Ver_zona.php	Vista de los productos	

MODULO BÚSQUEDA Y RECOMENDACIÓN DE PRODUCTOS TURÍTISTICOS

Usuario Final(Turista)

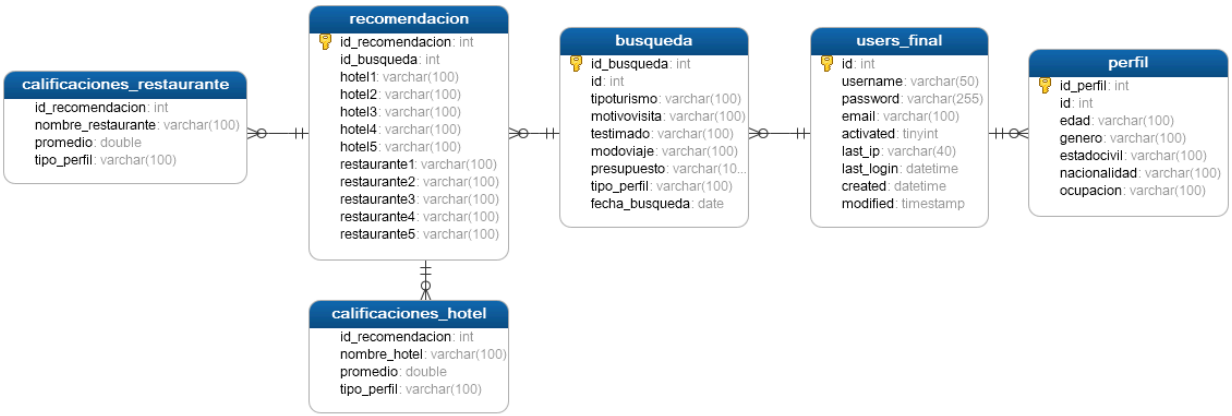


Figura 27. Búsqueda

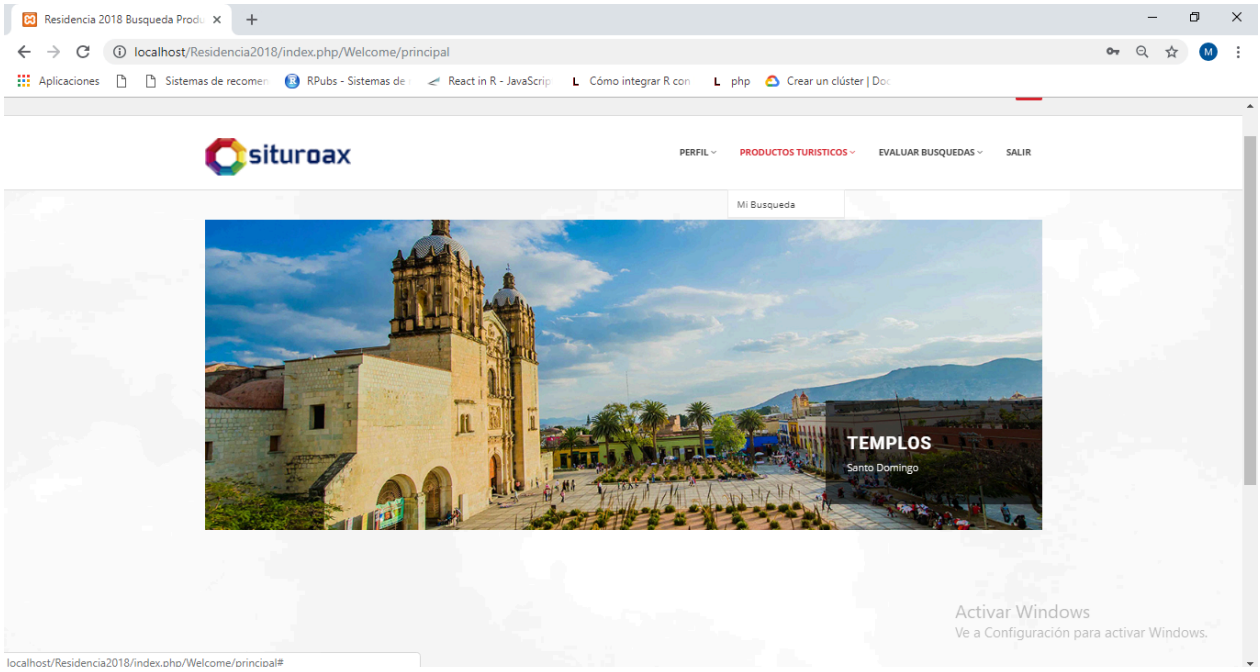


Figura 28. Pantalla búsqueda

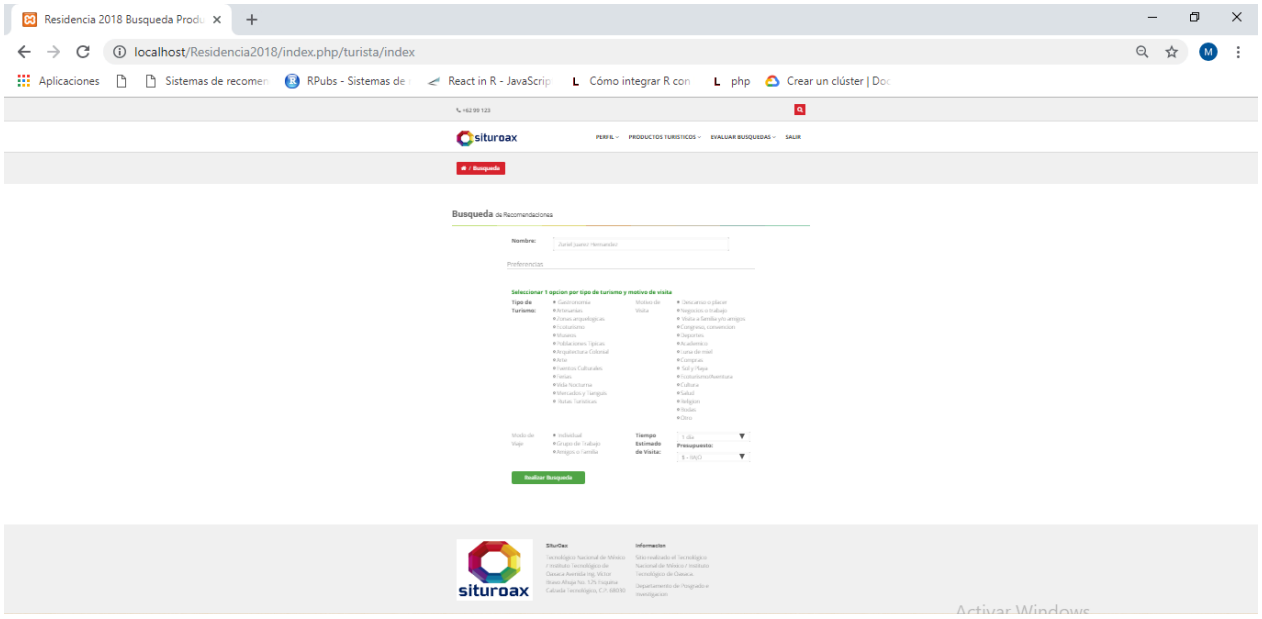


Figura 29. Prueba búsqueda

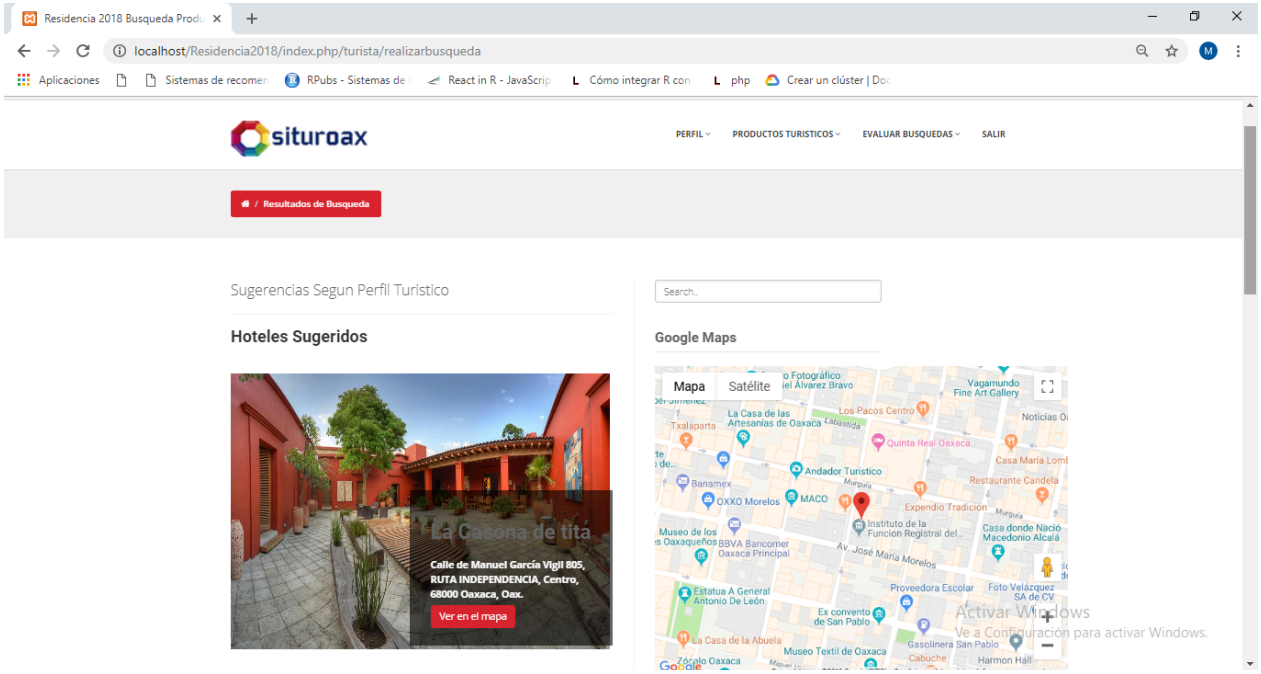


Figura 30. Mapa Resultados

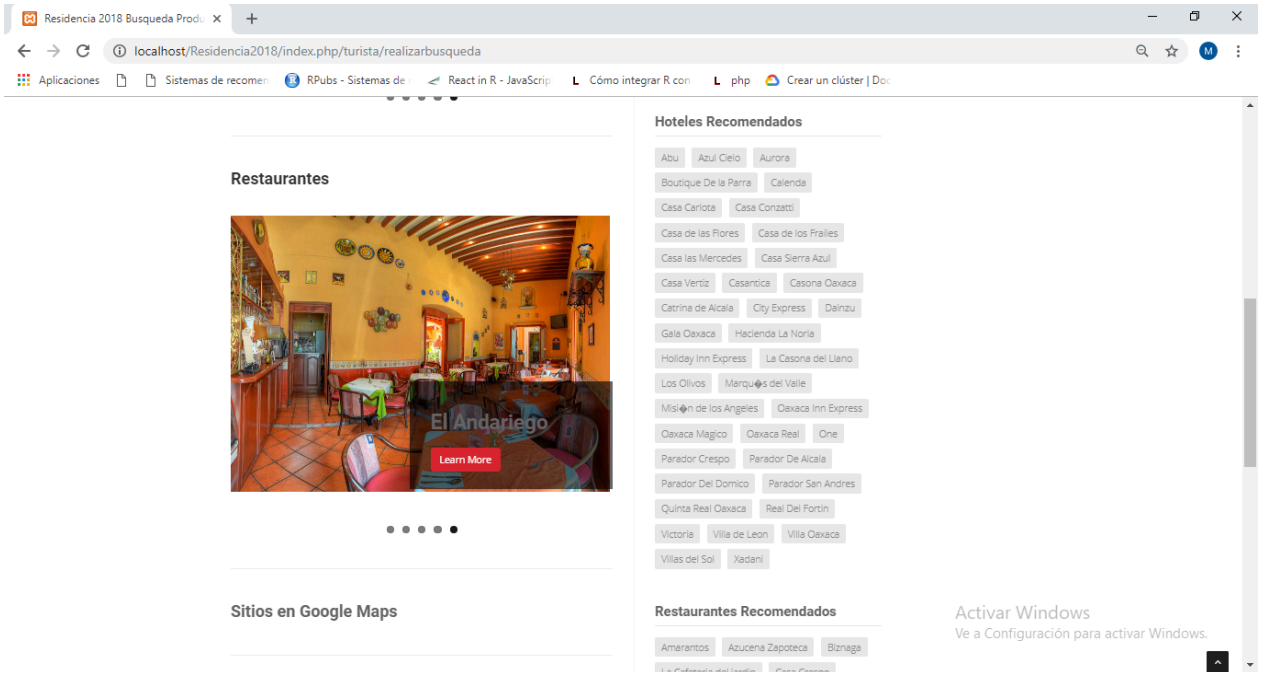


Figura 31.Recomendación Hotel

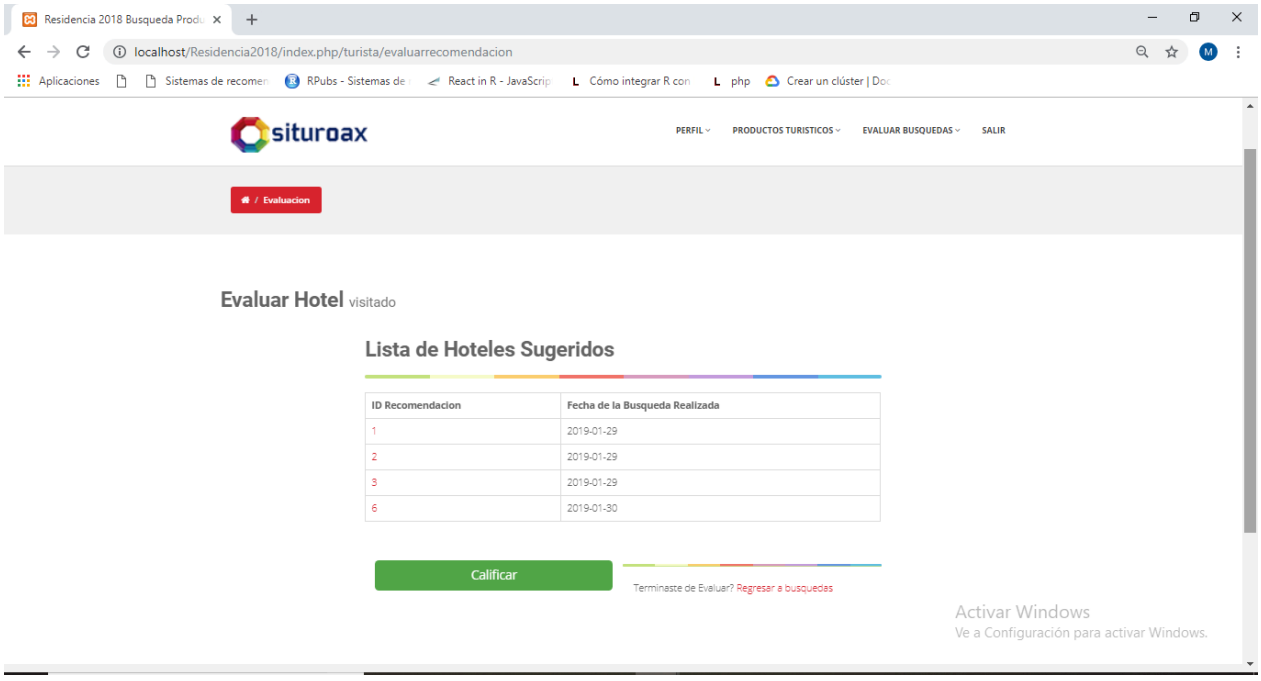


Figura 32.Evaluar Hotel

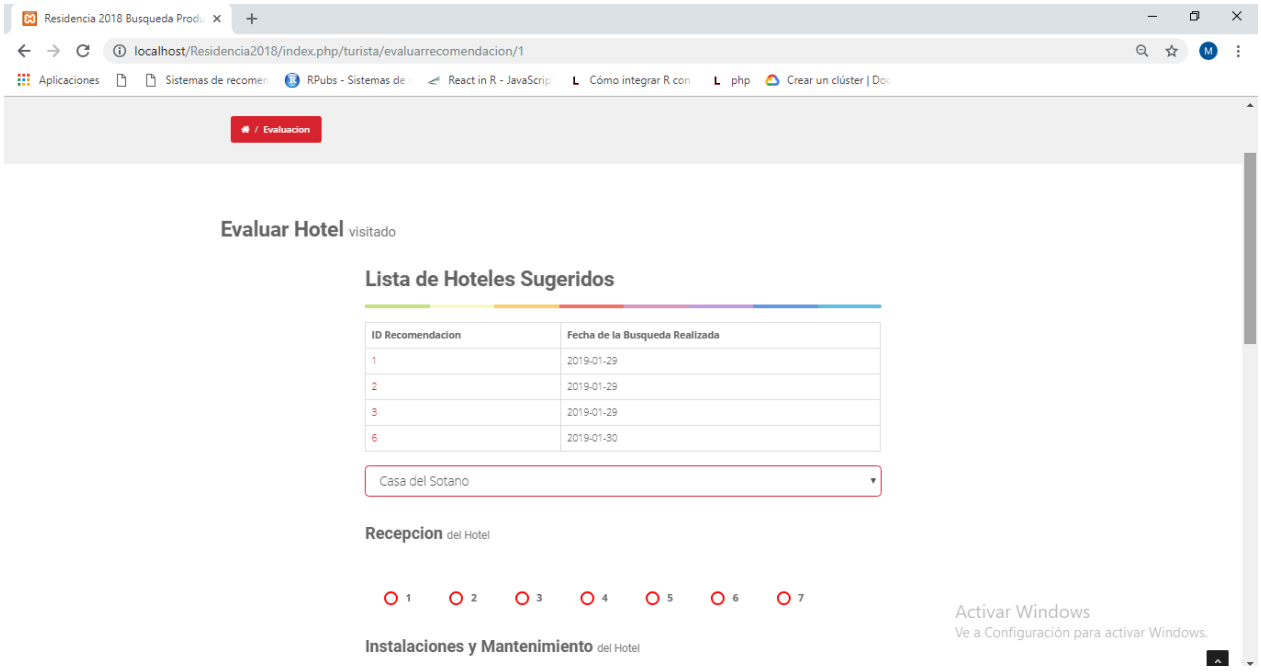


Figura 33. Evaluar Hotel Selcionado

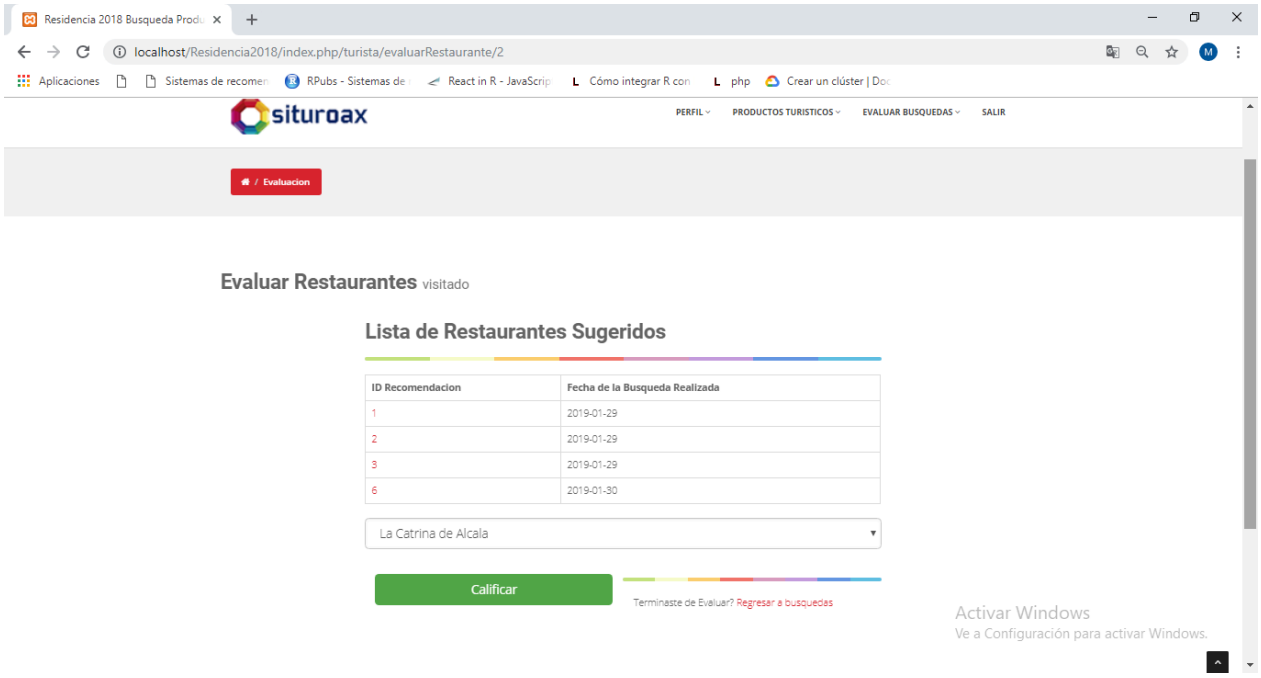


Figura 34. Evaluar Restaurante

Descripción de las clases del módulo búsquedas y recomendación de productos turísticos

Controladores

Ilustración 60 Controladores

nombre	funciones	Descripción	Ubicación
Welcomefinal	function __construct()		
	index()	Redirección para la búsqueda de productos	
	principal()	Invoca a la vista	
	ver()	Muestra lista de usuarios	
	edit()	Edita permisos	
	verperfil()	Muestra un perfil	
	editare()	Formulario para editar	
	actualizar_datos()	Edita el perfil	
	register()	Un nuevo registro	
turista			
	verperfil()	Muestra un perfil	
	verperfildatos()	Muestra los datos del perfil	
	index()	Muestra el id y las calificaciones del producto	
	realizarbusqueda()	Realiza la búsqueda de recomendaciones de hoteles y restaurantes y guarda las búsquedas	
	ImportCSV2Array($filename)	Importa csv para manipularlo en R	
	realizarevaluacion()	Importa el csv de calificaciones	
	evaluarrecomendacion()	Muestra los hoteles a evaluar	
	evaluarRestaurante()	Muestra los restaurantes a evaluar	
	calificar()	Evalúa hoteles	
	calificarR()	Evalúa restaurantes	