

Universidad de las Ciencias Informáticas



Facultad 2

Módulo destinos turísticos del Sistema de reseña, calificación y reservación de destinos turísticos

**Trabajo de diploma para optar por el Título de Ingeniero en
Ciencias Informáticas**

Autor: Luis Rodolfo Rodríguez Betancourt

Tutores:

MSc. Joaquín Quintas Santiago

MSc. Madelis Pérez Gil

La Habana, 2019

“Año 61 de la Revolución”

Sí aún no has encontrado algo que te guste hacer, sigue buscando. El trabajo va a ocupar gran parte de tu vida, de modo que asegúrate de vivir satisfecho de lo que haces.

Steve Jobs

Declaración de Autoría

Declaramos ser los únicos autores de la presente investigación y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Luis Rodolfo Rodríguez Betancourt

Firma Autor

Madelis Pérez Gil

Firma Tutor

Joaquín Quintas Santiago

Firma Tutor

Dedicatoria

Quiero dedicarles este momento a mi mamá y a mi papá por estar siempre preocupados y apoyándome durante todos estos años de estudio y esfuerzo.

Agradecimientos

Agradecerle primeramente a mi familia, especialmente a mis padres que son mi fuente de inspiración, por estar siempre preocupados y apoyándome durante estos años de estudio. Gracias por haber estado conmigo en todo momento.

A mis amigos **Luis Gabriel, Daniela, Beatriz** (Bia como todo el mundo le dice), y agradecerle especialmente a **Yisell** que más que una amiga se convirtió en mi hermana, a pesar de que pasamos muchos apuros nos ayudamos mutuamente durante todos estos años.

Agradecerles también a mis tutores **Madelis Pérez Gil** y **Joaquín Quintas Santiago** por la dedicación, ayuda y paciencia que me otorgaron. Este trabajo no hubiera sido posible sin ustedes.

A la **Universidad de las Ciencias Informáticas** y a todos los profesores que de alguna u otra forma contribuyeron a mi formación.

De manera general, agradecerle a todo aquel que de alguna forma me mostró su apoyo durante todos estos años. Gracias.

Resumen

La informática tiene su área de aplicación en la medicina, la educación y muchos otros sectores de la sociedad. En la Universidad de las Ciencias Informáticas se fomenta el soporte a la industria cubana del software por lo que brinda su aporte a la Empresa de Tecnologías de la Información para la Defensa, que está dedicada al desarrollo del software. La empresa incorpora una nueva forma de desarrollar software mediante el uso de la plataforma D'Prisa utilizando la arquitectura de microservicios, como estrategia para incursionar en el sector del turismo por proveerle mayores ganancias. Este sector necesitaba un sistema informático que almacene todos los destinos turísticos de Cuba sin importar el tipo, por esta razón se hizo necesario un sistema que permitió almacenar la información de cada uno de los destinos existentes en Cuba. El objetivo de la investigación es desarrollar un módulo informático que permita gestionar la información de todos los destinos turísticos en Cuba para el Sistema de reseña, calificación y reservación de destinos turísticos. Para la implementación del mismo se empleó el lenguaje JavaScript utilizando la herramienta Visual Studio Code respetando las fases de la metodología de desarrollo de software XP. Al concluir la investigación se obtuvo como resultado un módulo destinos turísticos del Sistema de reseña, calificación y reservación de destinos turísticos verificando la correspondencia de esta con los requisitos del cliente.

Palabras claves: almacenamiento de información, destino turístico, microservicios.

Índice general

Introducción	1
Capítulo 1. Fundamentación teórica	4
1.1 Introducción	4
1.2 Conceptos asociados	4
1.3 Estudio del estado del arte	5
1.3.1 Internacional	6
1.3.2 Nacional	9
1.3.3 Conclusiones del estudio del estado del arte	13
1.4 Metodología de desarrollo de software	14
1.4.1 Programación Extrema XP	16
1.5 Herramientas y lenguajes de programación	18
1.5.1 Lenguaje de Marcado de Hipertexto (HTML) 5	18
1.5.2 Hoja de Estilo en Cascada (CSS) 3	19
1.5.3 JavaScript	20
1.5.4 Framework CSS	20
1.5.5 Ajax	21
1.5.6 jQuery v3.3.1	21
1.5.7 Apache	22
1.5.8 Node.js v10.15.0	22
1.5.9 Sistema gestor de base de datos	23
1.5.10 Mocha v6.1.4	25
1.5.11 Herramienta CASE	25
1.5.12 Visual Studio Code v1.31.0	26
1.6 Conclusiones del capítulo	26
Capítulo 2. Exploración, Planificación y Diseño	27
2.1 Introducción	27
2.2 Descripción del módulo a desarrollar	27
2.2.1. Requisitos funcionales del módulo	27
2.2.2. Características del módulo informático	28
2.3 Exploración	29
2.3.1 Historias de usuario	29
2.4 Planificación	35
2.4.1 Estimación del esfuerzo	35
2.4.2 Plan de Iteraciones	36
2.4.3 Plan de entrega	37

2.5	Diseño	38
2.5.1	Patrones arquitectónicos	38
2.5.2	Patrones de diseño.....	39
2.5.3	Modelo de datos	40
2.5.4	Tarjetas Clase-Responsabilidad-Colaboración (CRC)	41
2.6	Conclusiones del capítulo	43
Capítulo 3.	Implementación y pruebas.....	44
3.1	Introducción	44
3.2	Implementación	44
3.2.1	Tareas de ingeniería.....	44
3.2.2	Pruebas de software.....	55
3.2.3	Pruebas unitarias.....	56
3.2.4	Pruebas de aceptación	57
3.2.5	Pruebas de integración.....	61
3.2.6	Índice de satisfacción grupal.....	63
3.3	Conclusiones del capítulo	65
Conclusiones	66
Recomendaciones	67
Bibliografía.....		68
Anexos		73
Anexo 1.	Entrevista	73
Anexo 2.	Encuesta de satisfacción de usuarios potenciales	73
Anexo 3.	Acta de aceptación	74

Índice de figuras

Figura 1: Pantalla principal del sistema informático Booking.com (Booking.com, 2018)	6
Figura 2: Pantalla principal del sistema informático Atrápalo (Atrápalo, 2018)	7
Figura 3: Pantalla principal del sistema informático Trivago (trivago, 2018)	8
Figura 4: Pantalla principal del sistema informático TripAdvisor (tripadvisor, 2018)	9
Figura 5: Pantalla principal del sistema informático Gaviota (gaviota, 2018)	10
Figura 6: Pantalla principal del sistema informático Cubanacan (cubanacan, 2018) ...	11
Figura 7: Pantalla principal del sistema informático Gran Caribe (gran-caribe, 2018) .	12
Figura 8: Pantalla principal del sistema informático Islazul (islazul, 2018).....	13
Figura 9: Modelo de datos del módulo informático (Elaboración propia)	41
Figura 10: Resultado de la ejecución de las pruebas unitarias.....	57
Figura 11: Resultados de la ejecución de las pruebas de aceptación	61
Figura 12: Captura de pantalla de la creación de los microservicios en la plataforma D'Prisa	62
Figura 13: Captura de pantalla del módulo integrado en la plataforma D'Prisa	62
Figura 14: Acta de aceptación	74

Índice de tablas

Tabla 1: Comparación entre los sistemas informáticos de almacenamiento de información de destinos turísticos y sus funcionalidades	14
Tabla 2: Historia de usuario #1	30
Tabla 3: Historia de usuario #2	30
Tabla 4: Historia de usuario #3	31
Tabla 5: Historia de usuario #4	32
Tabla 6: Historia de usuario #5	32
Tabla 7: Historia de usuario #6	33
Tabla 8: Historia de usuario #7	33
Tabla 9: Historia de usuario #8	34
Tabla 10: Historia de usuario #9	35
Tabla 11: Estimación de esfuerzo	36
Tabla 12: Plan de duración de las iteraciones.....	37
Tabla 13: Fechas de entrega	38
Tabla 14: Tarjeta CRC #1	42
Tabla 15: Tarjeta CRC #2	42
Tabla 16: Tarjeta CRC #3	42
Tabla 17: Tarjeta CRC #4	42
Tabla 18: Tarea de ingeniería # 1	45
Tabla 19: Tarea de ingeniería # 2	45
Tabla 20: Tarea de ingeniería # 3	48
Tabla 21: Tarea de ingeniería # 4	48
Tabla 22: Tarea de ingeniería # 5	49
Tabla 23: Tarea de ingeniería # 6	49
Tabla 24: Tarea de ingeniería # 7	51
Tabla 25: Tarea de ingeniería # 8	51
Tabla 26: Tarea de ingeniería # 9	52
Tabla 27: Tarea de ingeniería # 10	53
Tabla 28: Tarea de ingeniería # 11	53
Tabla 29: Tarea de ingeniería # 12	53
Tabla 30: Tarea de ingeniería # 13	54
Tabla 31: Tarea de ingeniería # 14	54
Tabla 32: Tarea de ingeniería # 15	55
Tabla 33: Caso de prueba de aceptación # 1	57
Tabla 34: Caso de prueba de aceptación # 2.....	58

Tabla 35: Caso de prueba de aceptación # 3.....	58
Tabla 36: Caso de prueba de aceptación # 4.....	58
Tabla 37: Caso de prueba de aceptación # 5.....	59
Tabla 38: Caso de prueba de aceptación # 6.....	59
Tabla 39: Caso de prueba de aceptación # 7.....	59
Tabla 40: Caso de prueba de aceptación # 8.....	60
Tabla 41: Caso de prueba de aceptación # 9.....	60
Tabla 42: Cuadro lógico de ladov (Elaboración propia).....	63
Tabla 43: Encuesta aplicada a los trabajadores de la XETID.....	73

Introducción

La informática a nivel mundial ha permitido que el hombre pueda potenciar sus capacidades de pensamiento, memoria y comunicación. Su área de aplicación no tiene límites; la misma puede ser utilizada en la gestión de negocios, en las comunicaciones, en el transporte, en la medicina, en la educación y en muchos otros sectores de la sociedad.

En Cuba, se está desarrollando el proceso de informatización de la sociedad. Muestra de ello es el uso del servicio de 3G para la telefonía celular y la apertura en todo el país de 154 salas de navegación públicas en el año 2018, permitiendo llegar a la comercialización del acceso de Banda Ancha (acceso a la red con mayor rapidez y prestaciones), en lo que se trabaja en la actualidad. A eso se debe que Cuba cuente con casi tres millones de usuarios con acceso a las redes, permitiendo que las personas puedan acceder a plataformas institucionales, sitios de prensa, redes sociales, entre otros. Este proceso ha evidenciado resultados satisfactorios en diversos sectores del país, como es el caso del sector del turismo, permitiendo tratar de forma digital la información asociada a los destinos turísticos de Cuba mediante ordenadores. De esta forma las personas pueden obtener información sobre las ofertas y servicios que se brindan en cada destino turístico (MINCOM, 2018).

La Empresa de Tecnologías de la Información para la Defensa (XETID) radica dentro de la infraestructura de la Universidad de las Ciencias Informáticas. La misma está dedicada al sector del software, la automática y las comunicaciones. Actualmente la XETID tiene como estrategia para aumentar sus ingresos, vincularse al desarrollo de software para el sector del turismo por ser este uno de los que más aporta a la economía cubana. Para ello ha identificado algunos elementos que pudieran repercutir de forma positiva en este sector (XETID, 2018). A continuación se mencionan:

- Lograr el almacenamiento de la información de todos los destinos turísticos de Cuba en un solo sistema.
- Lograr que cada responsable de los destinos turísticos pueda administrar la información de su destino turístico.

La XETID necesita una herramienta informática para el sector del turismo que permita a las personas almacenar la información correspondiente de su propio destino turístico, así como modificarla según sea necesario.

Descrita la situación problemática anterior se llega al siguiente **problema a resolver**:

¿Cómo contribuir a la gestión de la información de todos los destinos turísticos en Cuba para el Sistema de reseña, calificación y reservación de destinos turísticos?

Siendo el **objeto de estudio** de la investigación: Sistemas informáticos de gestión de información de destinos turísticos.

Para resolver el problema anterior, se establece como **objetivo general** de la investigación: Desarrollar un módulo informático que permita gestionar la información de todos los destinos turísticos en Cuba para el Sistema de reseña, calificación y reservación de destinos turísticos.

Se define como **campo de acción**: Sistemas informáticos de gestión de información de destinos turísticos utilizando microservicios.

Para guiar el desarrollo de la investigación, surgen las siguientes **preguntas científicas**:

- ¿Cuáles son los elementos teóricos y metodológicos para el desarrollo del módulo destinos turísticos del Sistema de reseña, calificación y reservación de destinos turísticos?
- ¿Cuál es el estado actual de los sistemas homólogos para conocer aspectos regulares para el diseño del módulo informático a desarrollar?
- ¿Cómo contribuir al diseño del módulo informático siguiendo la metodología de desarrollo de software adecuada?
- ¿Cómo los elementos anteriormente definidos favorecen al desarrollo del módulo informático esperado?
- ¿Cómo las pruebas que se le realizarán al sistema garantizarán el correcto funcionamiento del módulo informático?

Con la intención de darle cumplimiento a lo anteriormente planteado, se elaboran las siguientes **tareas de investigación**:

- Revisar la información recopilada para elaborar el marco teórico conceptual relacionado con el módulo, para el Sistema de reseña, calificación y reservación de destinos turísticos.
- Estudiar sistemas homólogos para conocer aspectos regulares sobre el diseño del módulo a implementar.

- Seleccionar las herramientas informáticas y metodologías de desarrollo de software existentes para realizar la implementación del módulo.
- Identificar las principales funcionalidades del módulo informático para su diseño e implementación, teniendo en cuenta los elementos anteriormente descritos.
- Realizar los diferentes tipos de pruebas de software, para garantizar el correcto funcionamiento del módulo.

Métodos científicos de la investigación

Los métodos científicos de la investigación son la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. Se clasifican en teóricos y empíricos, los cuales están dialécticamente relacionados (Coello González, y otros, 2012).

Método de investigación teórico:

- **Analítico-Sintético:** Este método se aplica para realizar un análisis sobre la información recopilada en el sector del turismo, asociada al proceso de almacenamiento de información de destinos turísticos en Cuba. Además, permite un análisis de la información actual a nivel nacional e internacional sobre Sistemas Informáticos (SI) similares. De esta forma se identificaron una serie de conceptos y otros elementos asociados para la presente investigación.

Método empírico:

- **Entrevista (Estructurada):** Se aplicó este método a Joaquín Quintas Santiago, cliente de la XETID. Con la aplicación de este método se recopilaron las funcionalidades que el cliente desea que estén presentes en el módulo a desarrollar. Además se recopiló información sobre cómo funciona la plataforma sobre la cual va a funcionar el módulo una vez concluido ([Ver anexo 1](#)).

Capítulo 1. Fundamentación teórica

1.1 Introducción

Este capítulo está enfocado a mostrar una descripción sobre todo lo relacionado a los sistemas de reseña, calificación y reservación de destinos turísticos y sus principales conceptos. Se realizará además un estudio del estado del arte tanto internacional como nacional. Por último se realizará un estudio para seleccionar la metodología de desarrollo de software adecuada, las herramientas informáticas de desarrollo y el lenguaje de programación a utilizar durante la implementación del módulo en cuestión.

1.2 Conceptos asociados

Sistema informático: Es un conjunto de partes o recursos formados por el hardware, software y las personas que lo emplean, que se relacionan entre sí para almacenar y procesar información con un objetivo en común (Alegsa, 2018).

Almacenar: Poner o guardar en almacén. Reunir, guardar o registrar en cantidad algo (Real Academia Española, 2018).

Según las definiciones de sistema informático y almacenar, el autor de la investigación decide definir un **sistema informático de almacenamiento de información** como: Sistema informático capaz de permitirle al usuario efectuar, de forma precisa, el proceso de almacenar información en un ordenador con un propósito determinado.

Destino turístico: Es la suma de los distintos productos turísticos ofrecidos por una determinada región geográfica delimitada, que al mismo tiempo se complementan y compiten entre sí. Los bienes y servicios, los recursos, y las infraestructuras y equipamientos, generalmente, son compartidos por los distintos productos turísticos de un mismo destino. Para que un destino turístico pueda ser considerado como tal y no simplemente como la suma de productos aislados, es fundamental una gestión y una imagen de marca conjunta de dicho destino (García, 2010).

Microservicios: Los microservicios son componentes de software que permiten implementar un sistema mediante módulos, cada uno con un propósito diferente y específico, deben ser lo suficientemente pequeños de tal manera que sean fáciles de mantener. Se sugiere que los microservicios deben de ser guiados por los límites del dominio haciendo uso del patrón *Model Driven Design* y que se define como un

enfoque, con necesidades complejas, que se fundamenta con una profunda conexión entre la implementación del software y el modelo del núcleo del negocio, el cual provee prácticas y terminologías para la toma de decisiones que tienen como propósito acelerar proyectos de software con modelos complejos. Los microservicios deben de funcionar como entidades separadas y autónomas, su diseño debe seguir los lineamientos de alta cohesión y bajo acoplamiento, para ello un microservicio no deberá de exponer todo su funcionamiento ya que aumenta la tendencia al acoplamiento entre los consumidores dificultando el mantenimiento de la aplicación cuando se desea hacer una actualización (Newman, 2015).

Plataforma D'Prisa: La plataforma D'Prisa fue desarrollada en el año 2017 por el MSc. Joaquín Quintas Santiago y un grupo de trabajadores de la XETID. La misma tiene como objetivo principal el desarrollo rápido de aplicaciones a través de un espacio colaborativo donde los desarrolladores podrán compartir código listo para su empleo en la forma de módulos de software autónomos denominados "MicroServicios". Para ello, deben estar previamente integrados los microservicios correspondientes que van a permitir, una vez seleccionados, ejecutar la aplicación informática deseada. Mediante D'Prisa que es una de las tecnologías que se están utilizando en la XETID, se podrán crear aplicaciones distribuidas y escalables. La plataforma D'Prisa aplica la filosofía de "MicroFramework", término usado para referirse a un marco de aplicación web minimalista. De esta forma no se requerirá emplear un marco de trabajo completo para emplear un par de funciones sencillas. Se podrá seleccionar única y exclusivamente los elementos necesarios para desarrollar la aplicación (Quintas, 2018).

1.3 Estudio del estado del arte

En la actualidad existe una gran variedad de sistemas informáticos que permiten a las personas almacenar información sobre destinos turísticos, por tal motivo se hizo necesario realizar un estudio del estado del arte tanto internacional como nacional. En la medida en que estos estudios se realizaron, más elementos fueron tomados como punto de partida para la toma de decisiones. Todo esto se hizo con el fin de hacer balances sobre los sitios que más se asemejaban a la idea que se quería desarrollar. El estado del arte se posicionó como una particularidad de búsqueda de la investigación y contribuye a que el sistema a desarrollar sirva de aporte a la sociedad y pueda resolver la necesidad para lo cual fue creado. A continuación, se describen

algunos de los sistemas informáticos de mayor similitud al módulo informático que se quiere implementar para esta investigación.

1.3.1 Internacional

Después de efectuar un estudio sobre los sistemas informáticos similares, dedicados al almacenamiento de información de destinos turísticos; se escogieron 4 y se estudiaron las principales funcionalidades que brindan y otros aspectos de interés en correspondencia al módulo que se desea desarrollar. A continuación, se muestran ejemplos de estos 4 sistemas, según el ranking internacional (Travelholic, 2017):

Booking.com es un sistema informático que almacena la información de destinos turísticos de casi cualquier parte del mundo y se accede a través de la url: <https://www.booking.com>, las principales funcionalidades de interés que brinda este sistema son:

- Permite gestionar la información de los destinos turísticos.
- Permite buscar destinos turísticos.

Aspectos de interés:

- Fue creado en el año 1996.
- La sede principal se encuentra ubicada en Ámsterdam.

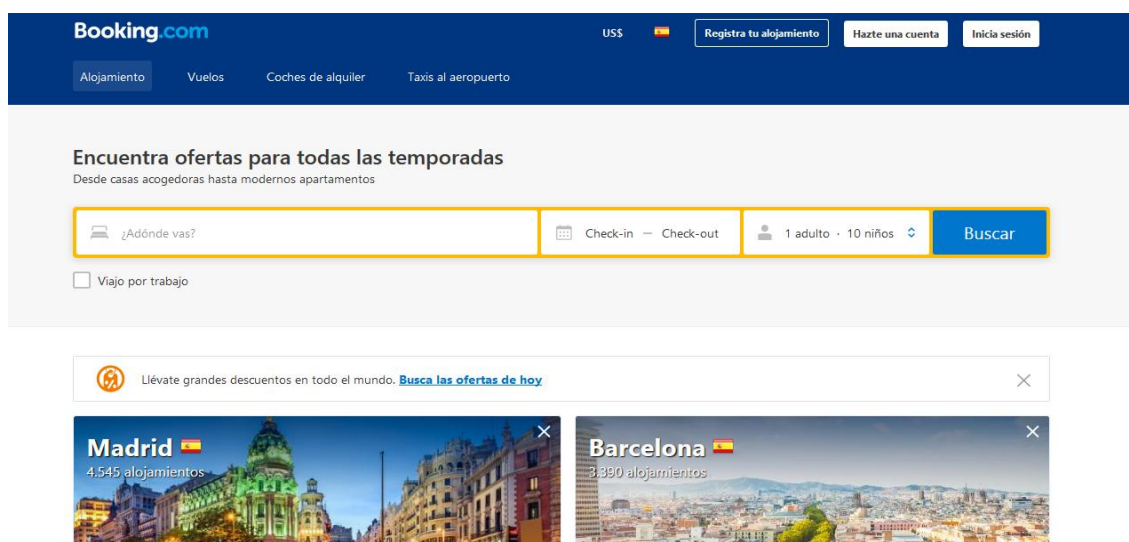


Figura 1: Pantalla principal del sistema informático Booking.com (Booking.com, 2018)

Atrápalo es un sistema informático que almacena la información de destinos turísticos de casi cualquier parte del mundo y se accede a través de la url:

<https://www.atrapalo.com/hoteles>, las funcionalidades de interés que brinda este sistema son:

- Permite gestionar información de los destinos turísticos.
- Permite buscar destinos turísticos.
- Muestra información sobre los servicios que se ofertan en cada hotel.

Aspecto de interés:

- Fue creado en el año 2000.



Figura 2: Pantalla principal del sistema informático Atrápalo (Atrápalo, 2018)

Trivago es un sistema informático que almacena la información de destinos turísticos de casi cualquier parte del mundo y se accede a través de la url: <https://www.trivago.es>. Las principales funcionalidades de interés que brinda este sistema son:

- Permite gestionar la información de los destinos turísticos.
- Permite buscar destinos turísticos.

Aspectos de interés:

- Fue creado en el año 2005.
- La sede principal se encuentra ubicada en Alemania.

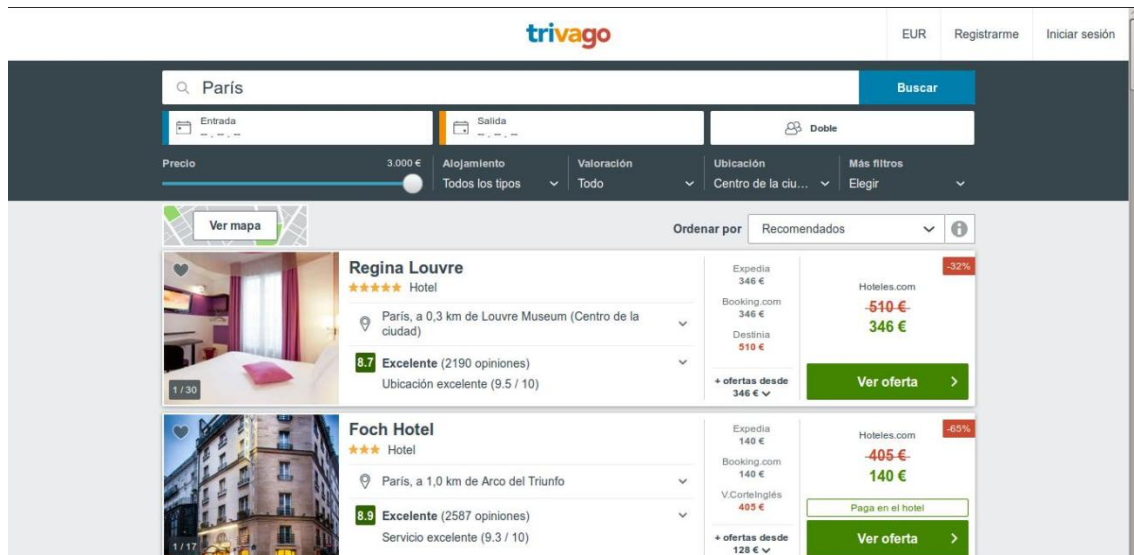


Figura 3: Pantalla principal del sistema informático Trivago (trivago, 2018)

TripAdvisor es un sistema informático que almacena la información de los destinos turísticos de casi cualquier parte del mundo y se accede a través de la url: <https://www.tripadvisor.es>. Las funcionalidades de interés que brinda este sistema son:

- Permite gestionar la información de los destinos turísticos.
- Permite buscar destinos turísticos.

Aspectos de interés:

- Fue creado en el año 2000.
- La sede principal se encuentra ubicada en EEUU.

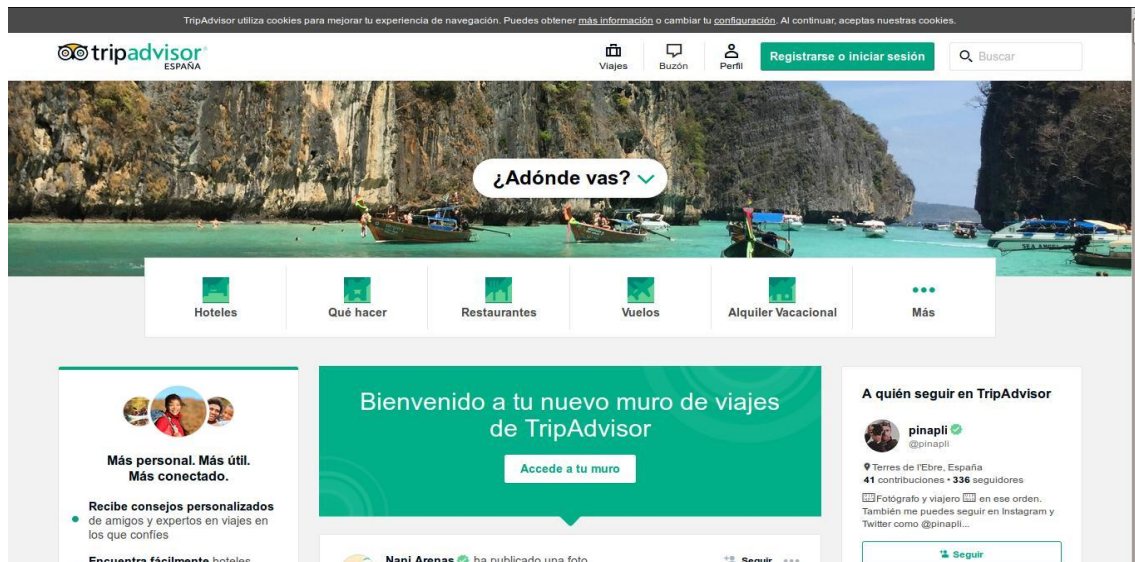


Figura 4: Pantalla principal del sistema informático TripAdvisor (tripadvisor, 2018)

1.3.2 Nacional

Después de efectuar un estudio sobre los sistemas informáticos de Cuba similares al módulo que se desea desarrollar, dedicados al almacenamiento de información de destinos turísticos; se escogieron 4 y se estudiaron las principales funcionalidades que brindan. A continuación se muestran 4 ejemplos de las principales cadenas hoteleras de Cuba (Cubazul, 2015):

Gaviota es un grupo hotelero que posee un portal web que almacena la información de los destinos turísticos de Cuba de su propio grupo hotelero. Se puede acceder a través de la url: <https://www.gaviota-grupo.com/es/hoteles>, las principales funcionalidades que brinda son:

- Permite gestionar información de destinos turísticos.
- Muestra una descripción sobre el hotel, las habitaciones, los restaurantes, los bares y los servicios que se ofrecen en los destinos turísticos.
- Permite buscar destinos turísticos.

Aspectos de interés:

- Fue creado en el año 1995.
- La sede principal se encuentra ubicada en La Habana (Cuba).

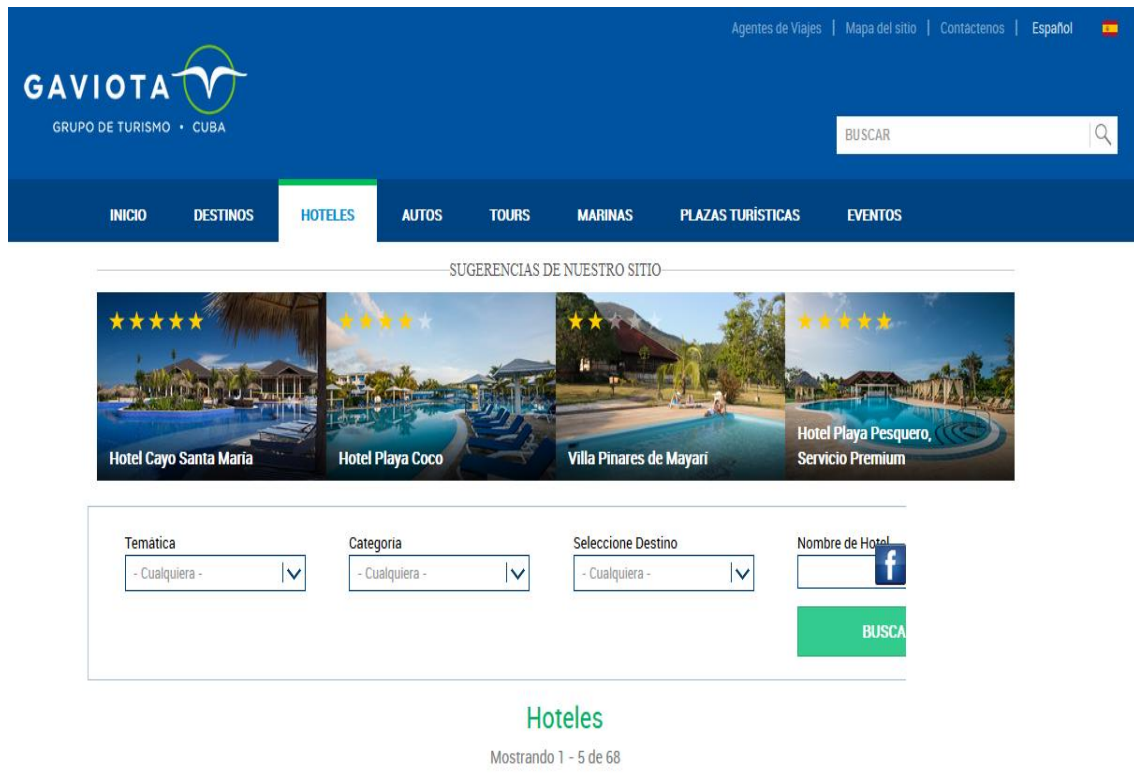


Figura 5: Pantalla principal del sistema informático Gaviota (gaviota, 2018)

Cubanacan es un grupo hotelero que posee un portal web que almacena la información de los destinos turísticos de Cuba de su propio grupo hotelero. Se puede acceder a través de la url: <https://www.hotelescubanacan.com>, las principales funcionalidades de interés que brinda son:

- Permite gestionar información de destinos turísticos.
- Muestra una descripción sobre el destino turístico.
- Permite buscar destinos turísticos.

Aspectos de interés:

- Fue creado en el año 1987.
- La sede principal se encuentra ubicada en Cuba.

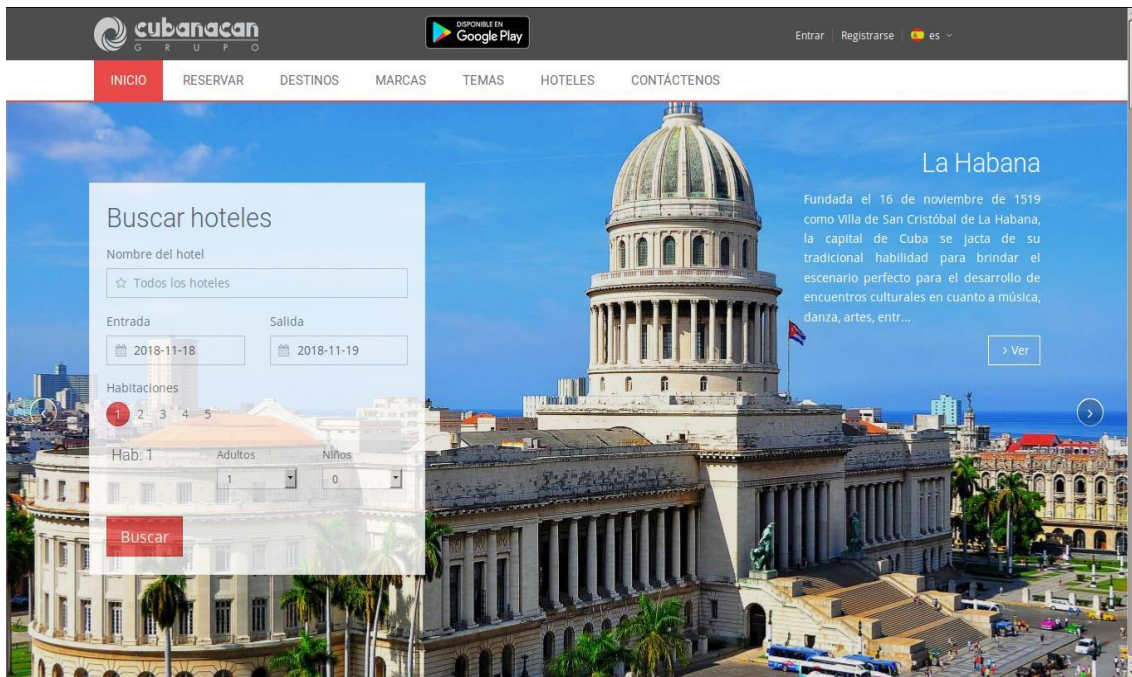


Figura 6: Pantalla principal del sistema informático Cubanacan (cubanacan, 2018)

Gran Caribe es un grupo hotelero que posee un portal web que almacena la información de los destinos turísticos de Cuba de su propio grupo hotelero. Se puede acceder a través de la siguiente url: <http://www.gran-caribe.cu>, las principales funcionalidades que brinda son:

- Permite gestionar información de destinos turísticos.
- Muestra una descripción sobre las habitaciones y los servicios que brinda el destino turístico.
- Permite buscar destinos turísticos.

Aspectos de interés:

- Fue creado en el año 1994.
- La sede principal se encuentra ubicada en La Habana (Cuba).

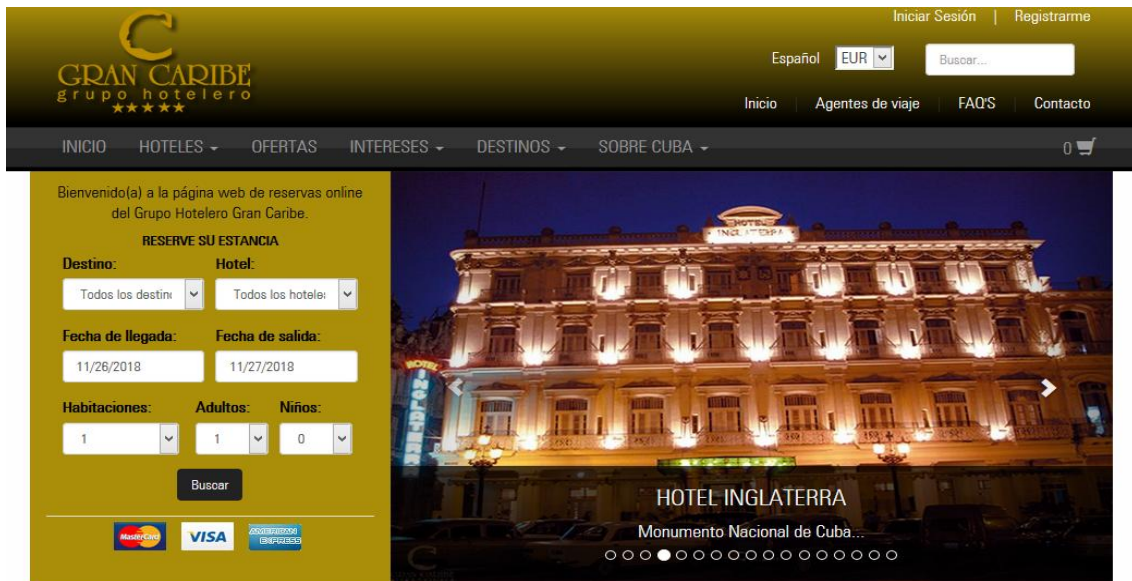


Figura 7: Pantalla principal del sistema informático Gran Caribe (gran-caribe, 2018)

La cadena hotelera Islazul posee un portal web que almacena la información de los destinos turísticos de Cuba que responden a la propia cadena hotelera. Se accede a través de la url: <http://islazul.cu>, las funcionalidades de interés que brinda son:

- Permite gestionar información de destinos turísticos.
- Permite buscar destinos turísticos.
- Brinda una descripción de los servicios que se ofrecen en los destinos turísticos.

Aspecto de interés:

- La sede principal se encuentra ubicada en Cuba.

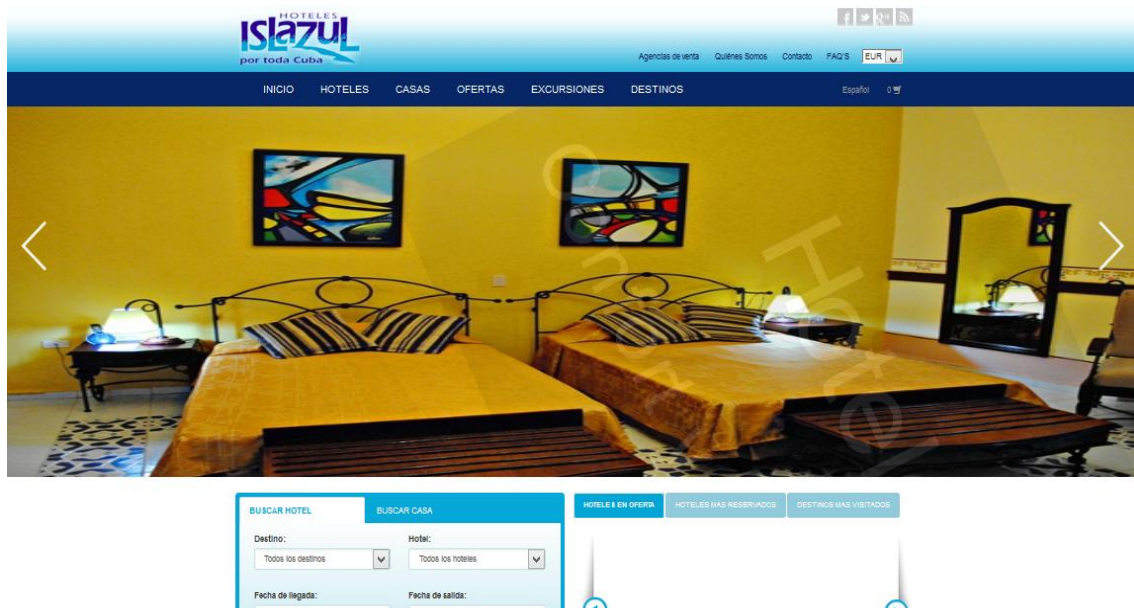


Figura 8: Pantalla principal del sistema informático Islazul (islazul, 2018)

1.3.3 Conclusiones del estudio del estado del arte

El resultado de realizar este estudio a los sistemas informáticos anteriores permitió identificar algunas de las funcionalidades que se utilizarán para la implementación del módulo. Estas son las que se aprecian en la parte del cliente puesto que no se puede acceder a la parte administrativa de los sistemas. A continuación se mencionan:

- Gestionar información de destinos turísticos.
- Buscar destinos turísticos.
- Mostrar la información sobre los servicios que se ofertan en cada destino turístico.

Después de analizar la información anterior, de forma general se identificaron los siguientes elementos concluyentes:

- Se identificaron las principales funcionalidades de los sistemas de almacenamiento de información de destinos turísticos tanto internacionales como nacionales.
- Los sistemas internacionales tienen destinos turísticos comunes, sin embargo, los nacionales solo permiten almacenar la información según el grupo hotelero al que responden.
- Los sistemas internacionales no tienen en cuenta todos los tipos de destinos turísticos existentes en Cuba.

- Tanto los sistemas internacionales como los nacionales no representan una solución a la problemática planteada en la XETID, puesto que no se puede acceder a su código fuente y no cuentan con todas las funcionalidades requeridas por el cliente.

Por ende la necesidad de crear este módulo que servirá de aporte al sector del turismo para poder permitirles a las personas poder almacenar la información de los destinos turísticos de Cuba. Para comprender de una mejor forma las funcionalidades que brindan cada uno de los sistemas informáticos de almacenamiento de información de destinos turísticos, tanto a nivel internacional como nacional, se realizó una tabla comparativa entre estos sistemas que evidencia las funcionalidades detectadas:

Tabla 1: Comparación entre los sistemas informáticos de almacenamiento de información de destinos turísticos y sus funcionalidades

Funcionalidades/Sistemas informáticos	Booking.com	Atrápalo	Trivago	TripAdvisor	Gaviota	Cubanacan	Gran Caribe	Islazul
Mostrar descripción de los servicios	No	Si	No	No	Si	Si	Si	Si
Buscar destinos turísticos	Si	Si	Si	Si	Si	Si	Si	Si
Gestionar información de destinos turísticos	Si	Si	Si	Si	Si	Si	Si	Si
Otros elementos a destacar/Sistemas informáticos								
Acceso al código fuente	No	No	No	No	No	No	No	No
Almacena todos los destinos turísticos cubanos	No	No	No	No	No	No	No	No

1.4 Metodología de desarrollo de software

La metodología de desarrollo de software es una de las etapas específicas de un trabajo o proyecto que parte de una posición teórica y conlleva a una selección de técnicas concretas o métodos acerca del procedimiento para el cumplimiento de los

objetivos. Determina los pasos a seguir y cómo realizarlos para finalizar una tarea. Estas se clasifican en dos grupos: tradicionales y ágiles (Maida, y otros, 2015).

Metodologías tradicionales

Las metodologías tradicionales centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto, definido todo esto, en la fase inicial del desarrollo del proyecto. Otra de las características importantes dentro de este enfoque son los altos costos al implementar un cambio y al no ofrecer una buena solución para proyectos donde el entorno es volátil. Las metodologías tradicionales se focalizan en documentación, planificación y procesos (Figueroa, y otros, 2008).

Metodología ágil

Las metodologías ágiles nacen como respuesta a los problemas de las metodologías tradicionales y se basa en dos aspectos puntuales: el retrasar las decisiones y la planificación adaptativa; permitiendo potenciar aún más el desarrollo de software a gran escala. Estas metodologías ponen de relevancia que la capacidad de respuesta a un cambio es más importante que el seguimiento estricto de un plan. Lo proponen porque para muchos clientes esta flexibilidad será una ventaja competitiva y porque estar preparados para el cambio significa reducir su costo (Figueroa, y otros, 2008).

Luego de analizados ambos grupos se decidió seguir una metodología ágil porque brinda rápida respuesta a los cambios, permite la intervención del cliente en el proceso, las entregas del producto por intervalos y la eliminación de tareas innecesarias. Todo esto propicia que la metodología ágil sea la indicada para el desarrollo del módulo en cuestión, permitiendo que se pueda realizar de la mejor manera y con el mínimo de dificultades posibles.

Selección de la metodología de desarrollo de software

Una revisión de las investigaciones, revisiones e implementaciones referenciadas por asociaciones como IEEE y ACM, arroja un tercer parámetro. Las metodologías más populares en los documentos científicos entre el 2003 y el 2007 fueron XP y Scrum. Sin embargo, cabe aclarar que la mayoría de lo que se publicó no está relacionado con una metodología en particular, sino con el concepto de metodologías ágiles en general; de hecho, el eje común a los temas tratados con mayor frecuencia es la presentación de experiencias exitosas de implantación. Otro criterio de selección es el reconocimiento de la alta adopción en la industria de desarrollo de software en los

últimos años. Según la séptima versión de la encuesta anual del estado del desarrollo de software usando metodologías ágiles, el tamaño promedio de las organizaciones es de 100 empleados. De quienes contestaron la encuesta, el 84% dijo que en sus compañías se practican las metodologías ágiles, lo que en algunos casos se ha venido haciendo por más de 5 años. La mayoría de equipos ágiles exitosos han adaptado prácticas ágiles de distintas metodologías para generar un proceso de desarrollo propio que se ajusta a sus necesidades. Estas adaptaciones parecen estar centradas en mezclas de Scrum con XP. La metodología XP se enfoca en prácticas de desarrollo mientras que Scrum apunta a la administración de proyectos (Metodologías Ágiles para el desarrollo de software: eXtreme Programming (XP), 2006).

1.4.1 Programación Extrema XP

De las metodologías ágiles, se decide escoger Programación Extrema o *Extreme Programming* (XP) porque es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Metodologías Ágiles para el desarrollo de software: eXtreme Programming (XP), 2006).

A continuación se muestran las ventajas de utilizar la metodología de desarrollo de software XP (Balarezo Penadillo, y otros, 2013):

- Ventajas:
 - ✓ Programación organizada
 - ✓ Menor tasa de errores
 - ✓ Satisfacción del programador
 - ✓ Solución de errores de programas
 - ✓ Versiones nuevas
 - ✓ Implementa una forma de trabajo donde se adapte fácilmente a las circunstancias.

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto, lo suficiente para que el diseño evolutivo funcione. Esto se consigue gracias a las tecnologías disponibles

para ayudar en el desarrollo de software y a la aplicación disciplinada de doce prácticas (Metodologías Ágiles para el desarrollo de software: eXtreme Programming (XP), 2006):

- **El juego de la planificación:** Hay una comunicación frecuente el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.
- **Entregas pequeñas:** Producir rápidamente versiones del sistema que sean operativas, aunque no cuenten con toda la funcionalidad del sistema. Esta versión ya constituye un resultado de valor para el negocio. Una entrega no debería tardar más de 3 meses.
- **Metáfora:** El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema (conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema, ayudando a la nomenclatura de clases y métodos del sistema).
- **Diseño simple:** Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto.
- **Pruebas:** La producción de código está dirigida por las pruebas unitarias; estas son establecidas por el cliente antes de escribirse el código y son ejecutadas constantemente ante cada modificación del sistema.
- **Refactorización (*Refactoring*):** Es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios. Se mejora la estructura interna del código sin alterar su comportamiento externo.
- **Programación en parejas:** Toda la producción de código debe realizarse con trabajo en parejas de programadores. Esto conlleva ventajas implícitas (menor tasa de errores, mejor diseño, mayor satisfacción de los programadores).
- **Propiedad colectiva del código:** Cualquier programador puede cambiar cualquier parte del código en cualquier momento.
- **Integración continua:** Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

- **40 horas por semana:** Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente esté ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo.
- **Cliente in-situ:** El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Este es uno de los principales factores de éxito del proyecto XP. El cliente conduce constantemente el trabajo hacia lo que aporta mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita.
- **Estándares de programación:** XP enfatiza que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible. El ciclo de vida ideal de XP es uno de los elementos más polémicos y sujetos a debate de la metodología. Mientras que Kent Beck, su fundador, la concibe como un conjunto de prácticas agrupadas en seis faces específicas, diversos autores difieren de su criterio, orientados a ajustar XP a las características propias de su entorno. La flexibilidad de esta metodología permite adaptarla a las necesidades de la organización. De esta forma, se pueden incorporar las prácticas que den resultado y modificar aquellas cuya especificación original no funcione de forma adecuada.

1.5 Herramientas y lenguajes de programación

Los lenguajes informáticos y las herramientas informáticas que se utilizarán en el desarrollo del módulo informático, están en correspondencia con el cliente ya que el mismo forma parte del equipo de desarrollo y solicitó que se realizara con las tecnologías y herramientas informáticas que se mencionan posteriormente, debido a que en un futuro se le proporcione soporte al módulo de la manera más factible posible.

1.5.1 Lenguaje de Marcado de Hipertexto (HTML) 5

El Lenguaje de Marcado de Hipertexto, en inglés, *Hypertext Mark-Up Language*, (HTML) es el lenguaje de marcado predominante para la construcción de páginas web. Permite representar el contenido enriquecido en forma de texto, así como complementar el texto con objetos, como el caso de las imágenes. HTML describe la estructura del contenido, además, puede manejar la apariencia de un documento y

también su comportamiento a través de un script, por ejemplo, JavaScript (Estándares de Diseño Web, 2010).

El lenguaje HTML pretende continuamente proporcionar funcionalidades más avanzadas para crear páginas más ricas en contenido. Así mismo se ha definido una especificación compatible con HTML: el XHTML (*Extensible Hypertext Markup Language*), que suele definirse como una versión XML de validación de HTML, proporciona un esquema XML que permita validar el documento para comprobar si está bien estructurado (Estándares de Diseño Web, 2010).

1.5.2 Hoja de Estilo en Cascada (CSS) 3

Las Hojas de Estilo en Cascada, del inglés, *Cascade Style Sheets* (CSS) es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar contenido y presentación; es imprescindible para crear aplicaciones web complejas. Separar contenido y presentación, brinda numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. El lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, el formato de tablas, la separación, el color, tamaño y tipo de letra de titulares y/o textos, la tabulación con la que se muestran los elementos de una lista o menú. Las ventajas de utilizar CSS son (Estándares de Diseño Web, 2010):

- Control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será empleada en una aplicación web, con lo que aumenta considerablemente la accesibilidad, por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o, incluso, a elección del usuario. Por ejemplo: para ser impresa, mostrada en un dispositivo móvil o ser leída por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se puede reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea).

1.5.3 JavaScript

JavaScript es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Como síntesis se puede decir que JavaScript es un lenguaje interpretado, basado en prototipos. Tiene como principal característica ser un lenguaje independiente de la plataforma. Es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con JavaScript se pueden crear diferentes efectos e interactuar con los usuarios. Este lenguaje posee varias características, entre ellas se puede mencionar (Valdés, 2007):

- Es un lenguaje basado en acciones que posee menos restricciones.
- Permite la programación orientada a objetos.
- Es soportado por la mayoría de los navegadores como Internet Explorer, Netscape, Opera, Mozilla Firefox, entre otros.

1.5.4 Framework CSS

En el desarrollo de software, un *framework* o “marco de trabajo”, es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto (Gutiérrez, et al., 2015).

Bootstrap v3.3.7 es un *framework* CSS desarrollado inicialmente (en el año 2011) por Twitter que permite dar forma a un sitio web mediante librerías CSS que incluyen tipografías, botones, cuadros, menús y otros elementos que pueden ser utilizados en cualquier sitio web. Aunque el desarrollo del *framework* Bootstrap fue iniciado por Twitter, fue liberado bajo licencia MIT en el año 2011 y su desarrollo continuo en un repositorio de GitHub. Bootstrap es una excelente herramienta para crear interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos y pantallas, sea cual sea su tamaño. Además, Bootstrap ofrece las herramientas necesarias para crear cualquier tipo de sitio web utilizando los estilos y elementos de sus librerías. Desde la aparición de Bootstrap 3 el *framework* se ha vuelto bastante más compatible con el

desarrollo web *responsive*, entre otras características se han reforzado las siguientes (Bootstrap, 2018):

- Soporte con HTML5 y CSS3, permitiendo ser usado de forma muy flexible para desarrollo web con unos excelentes resultados.
- Se ha añadido un sistema GRID que permite diseñar usando un GRID de 12 columnas donde se debe plasmar el contenido, con esto se puede desarrollar aplicaciones *responsive* de una forma mucho más fácil e intuitiva.
- Bootstrap 3 establece *Media Queries* para 4 tamaños de dispositivos diferentes, dependiendo del tamaño de su pantalla, estas *Media Queries* permiten desarrollar para dispositivos móviles y tablets de forma mucho más fácil.
- Bootstrap 3 también permite insertar imágenes *responsive*, es decir, con solo insertar la imagen con la clase “img-responsive” las imágenes se adaptaran al tamaño.

1.5.5 Ajax

AJAX no es una tecnología por sí misma, es un término que describe un nuevo modo de utilizar conjuntamente varias tecnologías existentes. Esto incluye: HTML o XHTML, CSS, JavaScript, DOM, XML, XSLT, y lo más importante, el objeto XMLHttpRequest. Cuando estas tecnologías se combinan en un modelo AJAX, es posible lograr aplicaciones web capaces de actualizarse continuamente sin tener que volver a cargar la página completa. Esto crea aplicaciones más rápidas y con mejor respuesta a las acciones del usuario (Ajax, 2018).

1.5.6 jQuery v3.3.1

jQuery es una librería de JavaScript rápida, pequeña y con muchas funciones. Permite que la manipulación de documentos HTML, el manejo de eventos, la animación y Ajax sean mucho más simples con una API fácil de usar que funciona en una gran cantidad de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript (jQuery, 2018).

El propósito de jQuery es facilitar el uso de JavaScript en su sitio web. jQuery toma muchas tareas comunes que requieren muchas líneas de código JavaScript para cumplirlas, y las envuelve en métodos a los que se pueden llamar con una sola línea de código. También simplifica muchas de las cosas complicadas de JavaScript, como las llamadas AJAX y la manipulación de DOM; además tiene complementos para casi

cualquier tarea y se ejecuta exactamente igual en todos los principales navegadores. La librería jQuery contiene las siguientes características (W3Schools.com, 2018):

- Manipulación HTML / DOM.
- Manipulación de CSS.
- Métodos de eventos HTML
- Efectos y animaciones
- AJAX
- Utilidades

1.5.7 Apache

Apache sigue siendo desarrollado por la comunidad de usuarios desarrolladores que trabaja bajo la tutela de *Apache Software Foundation*. Cuenta con el soporte necesario para tener páginas dinámicas. Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Posibilita configurar la creación y gestión de registros de actividad. Apache permite la creación de ficheros de registro a la medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor (Apache, 2018).

1.5.8 Node.js v10.15.0

Node.js es un entorno de programación en la capa del servidor basado en JavaScript, con entrada y salida de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Es una idea completamente revolucionaria ya que, a diferencia de las aplicaciones web tradicionales que usan PHP en el servidor, utiliza JavaScript, un modelo completamente asíncrono (Cid, 2014).

Los lenguajes del lado servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él. Para ello se utilizan tecnologías que garantizan esta interacción, como es el caso de Node.js. Node.js viene con muchos módulos útiles, de manera que no hay que escribir todo de cero (Kießling, et al., 2015).

Uno de los principales motivos por el cual se ha escogido Node.js es porque es especialmente útil cuando se van a realizar muchas operaciones simultáneas, sobretodo operaciones de entrada y salida. También es realmente bueno para aplicaciones en tiempo real, ya que necesitan una conexión persistente entre el cliente y el servidor.

1.5.9 Sistema gestor de base de datos

Un sistema gestor de bases de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener bases de datos, proporcionando acceso controlado a las mismas. Es una herramienta que sirve de interfaz entre el usuario y las bases de datos. Su objetivo fundamental consiste en suministrar al usuario las herramientas que le permitan manipular, en términos abstractos, los datos, o sea, de forma que no le sea necesario conocer el modelo de almacenamiento de los datos en la computadora, ni el método de acceso empleado (Galindo, 2011).

Bases de datos relacionales:

Es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), y a través de dichas conexiones relacionar los datos de ambas tablas, de ahí proviene su nombre: "Modelo Relacional". Las bases de datos relacionales poseen las siguientes características (Slideshare, 2014):

- Una base de datos se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y estas deben cumplir con la integridad de datos.
- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las formas relacionales.

Bases de datos no relacionales:

Las bases de datos no relacionales son las que, a diferencia de las relacionales, no tienen un identificador que sirva de relación entre un conjunto de datos y otros. La información se organiza normalmente mediante documentos y es muy útil cuando no se tiene un esquema exacto de lo que se va a almacenar. La indiscutible reina del reciente éxito de las bases de datos no relacionales es MongoDB seguida por Redis, Elasticsearch y Cassandra (Aukera, 2019).

Las características comunes entre todas las implementaciones de bases de datos distribuidas no relacionales, propietarias o no, suelen ser las siguientes (Slideshare, 2014):

- **Consistencia Eventual:** No se implementan mecanismos rígidos de consistencia como los presentes en las bases de datos relacionales, donde la confirmación de un cambio implica una comunicación del mismo a todos los nodos que lo repliquen. Esta flexibilidad hace que la consistencia se dé, eventualmente, cuando no se hayan modificado los datos durante un periodo de tiempo.
- **Estructura distribuida:** Generalmente se distribuyen los datos mediante mecanismos de tablas de hash distribuidas (DHT) ya que realmente se trata, según las distintas implementaciones, de redes p2p.
- **Escalabilidad horizontal:** La implementación típica se realiza en muchos nodos de capacidad de procesamiento limitado, en vez de utilizar grandes *mainframes*.
- **Tolerancia a fallos y Redundancia.**

No merece la pena ponerse a diseñar una base de datos relacional para almacenar esta información. En este caso, lo mejor es dejar a una base de datos no relacional que almacene todo lo que las máquinas reportan, tal cual. Por tanto, se selecciona el sistema gestor de base de datos no relacional para la administración de la Base de Datos del módulo a desarrollar.

MongoDB v4.0.2

MongoDB es una base de datos multiplataforma y orientada a documentos que proporciona alto rendimiento, alta disponibilidad y fácil escalabilidad. MongoDB trabaja en concepto de colección y documento. La base de datos es un contenedor físico para las colecciones. Cada base de datos obtiene su propio conjunto de archivos en el sistema de archivos. Un único servidor MongoDB generalmente tiene múltiples bases de datos. La colección es un grupo de documentos de MongoDB. Es el equivalente a una tabla de un sistema de gestión de bases de datos relacionales. Existe una colección dentro de una sola base de datos. Las colecciones no imponen un esquema. Los documentos dentro de una colección pueden tener diferentes campos. Normalmente, todos los documentos de una colección tienen un propósito similar o relacionado. Un documento es un conjunto de pares clave-valor. Los documentos tienen esquema dinámico. El esquema dinámico significa que los documentos en la

misma colección no necesitan tener el mismo conjunto de campos o estructura, y los campos comunes en los documentos de una colección pueden contener diferentes tipos de datos (tutorialspoint, 2019).

Ventajas de utilizar MongoDB (Rodríguez, 2014):

- Mongo DB tiene la capacidad de realizar consultas utilizando JavaScript, haciendo que estas sean enviadas directamente a la base de datos para ser ejecutada.
- Se utiliza un sistema de archivos, ya que cuenta con la capacidad para balancear la carga y recopilación de datos utilizando múltiples servidores para almacenamiento de archivo.
- El desarrollador elige una llave (clave).
- La configuración es automática, se pueden agregar nuevas máquinas a MongoDB con el sistema de base corriendo.

Estas características generales convierten a MongoDB en el mecanismo de base de datos más adecuado para utilizarse en el desarrollo del módulo informático en cuestión.

1.5.10 Mocha v6.1.4

Mocha es un marco de prueba de JavaScript con numerosas funciones que se ejecuta en Node.js y en el navegador, lo que hace que las pruebas asíncronas sean simples. Las pruebas de Mocha se ejecutan en serie, lo que permite informes flexibles y precisos, al tiempo que se asignan excepciones no detectadas a los casos de prueba con resultados satisfactorios (Mocha, 2019). Se considera utilizar Mocha utilizando la librería Chai para realizar las pruebas unitarias del módulo informático que se desea implementar. Chai es una biblioteca de aserciones, similar a la función incorporada de Node.js. Facilita las pruebas al brindarle muchas aserciones que puede ejecutar en contra de su código (npmjs, 2018).

1.5.11 Herramienta CASE

Con el objetivo de facilitar las tareas del desarrollo de software, surgen herramientas informáticas que agilizan la labor en la ingeniería de requisitos. Dichas herramientas son denominadas CASE (Ingeniería de software asistida por computador), y sirven de apoyo para los desarrolladores, desde el principio hasta el final del proceso (Herramientas CASE para la ingeniería de requisitos, 2008).

Visual Paradigm v8.0 sirve de apoyo para los programas informáticos. Esta herramienta ayuda con la codificación y el diseño del software. Permite generar el código fuente de los programas y la documentación. Esta herramienta soporta el ciclo de vida completo del proceso de desarrollo del software; el modelo y el código permanecen sincronizados en todo este ciclo de desarrollo. Propicia la representación de todo tipo de diagramas, es multiplataforma y permite la compatibilidad entre diferentes ediciones. Esta es una herramienta de software libre de gran utilidad para el analista de un proyecto. Su principal ventaja es que permite la generación de base de datos, transformación de diagramas de entidad relación en tablas de base de datos (visual-paradigm, 2018).

1.5.12 Visual Studio Code v1.31.0

Visual Studio Code es un editor de código, cuyas características se asemejan a otros como Geany o el más reciente Brackets de Adobe. Soporta una cantidad considerable de lenguajes, ya sean propios de Microsoft como C#, F# y Visual Basic, o de otros como PHP, Python, Perl, SQL, shell scripting en Bash y Java, siendo este último el gran rival de .NET. También soporta Git y programación web con HTML, CSS y JavaScript, entre otros lenguajes (muylinux, 2015).

1.6 Conclusiones del capítulo

Los conceptos asociados a la problemática presente en la investigación, enriquecieron el conocimiento sobre sistemas informáticos y los llamados microservicios. De igual forma, aunque los sistemas informáticos de almacenamiento de información de destinos turísticos estudiados no satisfacen las necesidades del cliente en su totalidad, si permitieron la selección de algunas funcionalidades importantes para el desarrollo del módulo en cuestión. Además, la selección de la metodología de desarrollo de software XP permitió guiar el proceso de desarrollo de software teniendo en cuenta las fases que define.

Capítulo 2. Exploración, Planificación y Diseño

2.1 Introducción

En este capítulo se hará referencia a las fases de exploración, planificación y diseño, que son fases propias de la metodología XP. Además, se generan una serie de artefactos que conllevan a la etapa de implementación del módulo: historias de usuario, plan de iteración y las tarjetas clase - responsabilidad - colaboradores.

2.2 Descripción del módulo a desarrollar

El módulo destinos turísticos del Sistema de reseña, calificación y reservación de destinos turísticos va a servir como herramienta para que las personas que sean responsables de los destinos turísticos de Cuba, puedan almacenar la información de su propio destino turístico. De igual modo va a permitir modificar la información almacenada según sea necesario. El módulo en cuestión será implementado con microservicios que serán integrados posteriormente a la plataforma D'Prisa; requisito solicitado por el cliente para su uso.

2.2.1. Requisitos funcionales del módulo

Se identificaron una serie de requisitos (R) a través de la entrevista realizada al cliente y los sistemas homólogos estudiados, donde fueron determinados un conjunto de necesidades que marcan puntos claves para el desarrollo de la aplicación.

R1: Autenticar usuario

R2: Gestionar datos de los destinos turísticos

R3: Gestionar datos de los usuarios

R4: Buscar destino turístico

R5: Buscar usuario

R6: Mostrar cantidad de destinos turísticos

R7: Mostrar cantidad de usuarios

R8: Mostrar vista previa del destino turístico

R9: Mostrar ayuda

2.2.2. Características del módulo informático

Las características (C) del módulo están en correspondencia con los llamados requisitos no funcionales establecidos por otras metodologías de desarrollo de software. Para entender mejor las funcionalidades que debe realizar el sistema se establecen las siguientes características del módulo, a continuación se mencionan:

De usabilidad:

C1: Las interfaces del módulo deben cumplir con las pautas de diseño establecidas en la XETID.

De rendimiento:

C2: Servidor de base de datos con microprocesador de 4 núcleos o superior, RAM de 16GB y disco duro de 1TB.

De Portabilidad:

C3: Para una computadora personal utilizar un navegador web Mozilla Firefox 47.x o superior o Google Chrome 50.x o superior.

C4: Para un dispositivo móvil utilizar un navegador web Mozilla Firefox 47.x o superior o Google Chrome 50.x o superior y Android v4.2.2 o superior.

De implementación:

C5: SGBD: MongoDB v4.0.2.

C6: Plataforma D'Prisa.

C7: Lenguaje de programación del lado del servidor: Node.js v10.15.0.

De seguridad:

C8: Garantizar el acceso a las funcionalidades definidas para los usuarios de acuerdo a los roles que posean.

2.3 Exploración

Es la fase en la que se define el alcance general del proyecto. En ella, el cliente define lo que necesita mediante la redacción de sencillas historias de usuarios y los programadores estiman los tiempos de desarrollo en base a esta información. Las estimaciones realizadas en esta fase son primarias (ya que estarán basadas en datos de muy alto nivel), y podrían variar cuando se analicen más en detalle en cada iteración. La fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado (Joskowicz, 2008).

2.3.1 Historias de usuario

Dentro de la fase de exploración de la metodología XP, se realizan las historias de usuario (HU), estas tienen el mismo propósito que los casos de uso. Las mismas son escritas por los propios clientes, tal y como ellos visualicen las necesidades del sistema. Las HU son similares al empleo de escenarios, con la excepción de que no se limitan a la descripción de la interfaz de usuario. También conducirán el proceso de creación de los casos de prueba de aceptación que son empleados para verificar que las HU han sido implementadas correctamente (Escribano, 2002).

Para llegar a un consenso de que HU es más importante resolver, según los requisitos exigidos por el cliente y poder efectuar la planificación del módulo que se desea desarrollar; cada HU se clasifica por su prioridad dada por el cliente en (alta, media, baja):

- Alta: HU que sea fundamental en la implementación del módulo.
- Media: HU que presenten funcionalidades importantes pero que no resulten imprescindibles para el cliente.
- Baja: HU de apoyo para aquellas funcionalidades que si son primordiales para el cliente.

Además se determinó que podrían existir errores durante la implementación de cada HU y la dificultad de su implementación, por lo que serán clasificadas según el riesgo en su implementación (alto, medio, bajo):

- Alto: cuando en el desarrollo de la HU existan errores que provoquen la inoperatividad de la codificación.
- Medio: cuando existan errores en el desarrollo de la HU que retrasen el proceso de entrega de la versión.

- Bajo: cuando existan errores que pueden ser resueltos fácilmente sin que le provoquen daños al código del módulo.

Luego del análisis realizado respecto a la solicitud encomendada por el cliente, todos los parámetros quedaron esclarecidos, propiciando que todas las HU pudieran ser elaboradas en la etapa de exploración, sin tener que realizar modificaciones ni adicionar nuevas HU en las próximas iteraciones. Los prototipos de las HU fueron elaborados en la herramienta Microsoft Paint.

Las HU definidas fueron:

Tabla 2: Historia de usuario #1

Historia de Usuario	
Número: 1	Nombre: Autenticar usuario
Usuario: Administrador, usuarios	
Prioridad en Negocio: Alta	Iteración Asignada: 1
Riesgo en Desarrollo: Alto	Puntos Estimados: 0.5
Descripción: El administrador y los usuarios se autentican en el sistema para gestionar los datos de los destinos turísticos.	
Observaciones: El administrador y los usuarios tienen que estar registrados en la base de datos del sistema.	
Prototipo de Interfaz:	
<p>Bienvenido al Sistema de reseña, calificación y reservación de destinos turísticos</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px; width: 80%;"> <p style="color: #ccc;">Usuario</p> </div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 10px; width: 80%;"> <p style="color: #ccc;">Contraseña</p> </div> <div style="text-align: right; margin-top: 10px;"> <div style="background-color: #008000; color: white; padding: 5px 15px; border: 1px solid black; display: inline-block;"> Acceder </div> </div> </div>	

Tabla 3: Historia de usuario #2

Historia de Usuario	
Número: 2	Nombre: Gestionar datos de los destinos turísticos

Usuario: Administrador, usuarios		
Prioridad en Negocio: Alta	Iteración Asignada: 1	
Riesgo en Desarrollo: Alto	Puntos Estimados: 2.5	
Descripción: El administrador crea, modifica, muestra y elimina los datos de los destinos turísticos según sea necesario. En el caso de los usuarios, crean, modifican, muestran y eliminan los datos del destino turístico que les corresponde.		
Observaciones: El administrador solo puede gestionar los datos que le suministra el proveedor de los destinos turísticos. Los usuarios solo pueden gestionar los datos de su propio destino turístico.		
Prototipo de Interfaz:		
<div style="border: 1px solid black; padding: 10px;"> <div style="background-color: #4CAF50; color: white; padding: 5px;">Crear destino turístico</div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 45%;"> <p>*Nombre</p> <input type="text"/> </div> <div style="width: 45%;"> <p>*Alojamiento</p> <input type="text" value="v"/> </div> </div> <div style="margin-top: 10px;"> <p>*Descripción</p> <input style="width: 100%;" type="text"/> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 45%;"> <p>*Precio (CUP)</p> <input type="text"/> </div> <div style="width: 45%;"> <p>*Provincia</p> <input type="text" value="v"/> </div> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 30%;"> <p>*Teléfono</p> <input type="text"/> </div> <div style="width: 30%;"> <p>Teléfono</p> <input type="text"/> </div> <div style="width: 30%;"> <p>*Correo</p> <input type="text"/> </div> </div> <div style="margin-top: 10px;"> <p>*Mapa</p> <input style="width: 100%;" type="text"/> </div> <div style="display: flex; justify-content: space-between; margin-top: 10px;"> <div style="width: 30%;"> <p>*Servicio</p> <input type="text"/> </div> <div style="width: 10%; text-align: center;"> <p>Añadir</p> <input style="background-color: #4CAF50; color: white; border: none; width: 30px; height: 30px; border-radius: 50%; text-align: center; line-height: 30px; margin: 0 auto 0 auto;"/>+</div> <div style="width: 30%;"> <p>*Listado de servicios</p> <input type="text" value="v"/> </div> </div> <div style="margin-top: 10px;"> <p>*Imagen de la portada</p> <input style="width: 100%;" type="text"/> </div> <div style="text-align: center; margin-top: 10px;"> <input style="background-color: #4CAF50; color: white; border: none; padding: 5px 15px; border-radius: 5px;" type="button" value="Examinar"/> </div> <div style="display: flex; justify-content: flex-end; margin-top: 20px; gap: 10px;"> <div style="background-color: #FF9800; color: white; padding: 5px 15px; border-radius: 5px; text-align: center;">Cancelar</div> <div style="background-color: #4CAF50; color: white; padding: 5px 15px; border-radius: 5px; text-align: center;">Aceptar</div> </div> </div>		

Tabla 4: Historia de usuario #3

Historia de Usuario	
Número: 3	Nombre: Gestionar datos de los usuarios
Usuario: Administrador	
Prioridad en Negocio: Media	Iteración Asignada: 2

Riesgo en Desarrollo: Alto	Puntos Estimados: 2.5										
Descripción: El administrador crea, modifica, muestra y elimina los datos de los usuarios que van a gestionar los destinos turísticos.											
Observaciones: -											
Prototipo de Interfaz:											
<div style="border: 1px solid black; padding: 10px;"> <div style="background-color: #4CAF50; color: white; padding: 5px;">Crear usuario</div> <table style="width: 100%; margin-top: 10px;"> <tr> <td style="width: 33%;">Nombre <input type="text"/></td> <td style="width: 33%;">Apellidos <input type="text"/></td> <td style="width: 33%;">Carnet de identidad <input type="text"/></td> </tr> <tr> <td>Usuario <input type="text"/></td> <td>Contraseña <input type="text"/></td> <td>Repetir contraseña <input type="text"/></td> </tr> <tr> <td>Correo <input type="text"/></td> <td colspan="2"></td> </tr> </table> <div style="text-align: right; margin-top: 10px;"> Cancelar Aceptar </div> </div>			Nombre <input type="text"/>	Apellidos <input type="text"/>	Carnet de identidad <input type="text"/>	Usuario <input type="text"/>	Contraseña <input type="text"/>	Repetir contraseña <input type="text"/>	Correo <input type="text"/>		
Nombre <input type="text"/>	Apellidos <input type="text"/>	Carnet de identidad <input type="text"/>									
Usuario <input type="text"/>	Contraseña <input type="text"/>	Repetir contraseña <input type="text"/>									
Correo <input type="text"/>											

Tabla 5: Historia de usuario #4

Historia de Usuario	
Número: 4	Nombre: Buscar destino turístico
Usuario: Administrador	
Prioridad en Negocio: Baja	Iteración Asignada: 3
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.4
Descripción: El administrador realiza la búsqueda de los destinos turísticos según el nombre de los mismos.	
Observaciones: -	
Prototipo de Interfaz:	
<div style="border: 1px solid black; padding: 10px;"> <div style="background-color: #4CAF50; color: white; padding: 5px;">Buscar destino turístico</div> <div style="margin-top: 10px;"> <input style="width: 80%; height: 30px;" type="text"/> Buscar </div> </div>	

Tabla 6: Historia de usuario #5

Historia de Usuario	
Número: 5	Nombre: Buscar usuario

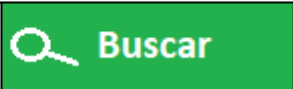
Usuario: Administrador	
Prioridad en Negocio: Baja	Iteración Asignada: 3
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.4
Descripción: El administrador realiza la búsqueda de las personas responsables según el usuario asignado a cada persona.	
Observaciones: -	
Prototipo de Interfaz:	
Buscar usuario	
<input type="text"/> 	

Tabla 7: Historia de usuario #6




Historia de Usuario						
Número: 6	Nombre: Mostrar cantidad de destinos turísticos					
Usuario: Administrador, usuarios						
Prioridad en Negocio: Baja	Iteración Asignada: 3					
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.5					
Descripción: El sistema muestra cuantos destinos turísticos hay almacenados.						
Observaciones: -						
Prototipo de Interfaz:						
Listado de destinos turísticos almacenados en el sistema						
Nombre	Alojamiento	Precio(CUP)	Provincia	Municipio	Responsable	Opciones
						  

Tabla 8: Historia de usuario #7

Historia de Usuario	
Número: 7	Nombre: Mostrar cantidad de usuarios
Usuario: Administrador	
Prioridad en Negocio: Baja	Iteración Asignada: 3
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.5
Descripción: El sistema muestra cuantos usuarios hay almacenados.	
Observaciones: -	

Prototipo de Interfaz:

Listado de usuarios almacenados en el sistema



Nombre	Apellidos	Carnet de identidad	Usuario	Correo	Opciones
					 

Tabla 9: Historia de usuario #8

Historia de Usuario

Número: 8

Nombre: Mostrar vista previa del destino turístico

Usuario: Administrador, usuarios

Prioridad en Negocio: Baja

Iteración Asignada: 3

Riesgo en Desarrollo: Bajo

Puntos Estimados: 1

Descripción: El sistema muestra una vista previa del destino turístico con todos los datos que fueron proporcionados por el administrador o los usuarios.

Observaciones: El sistema debe poseer al menos un destino turístico almacenado en la BD.

Prototipo de Interfaz:

Nombre del Hotel

☆☆☆☆☆
Hotel #estrellas

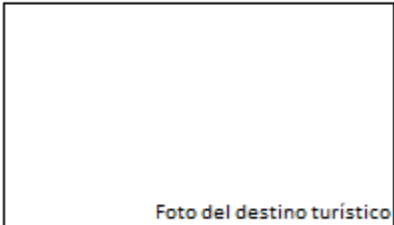
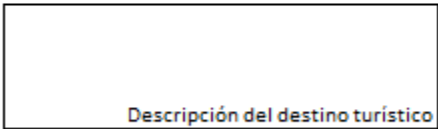


Foto del destino turístico

Descripción



Descripción del destino turístico

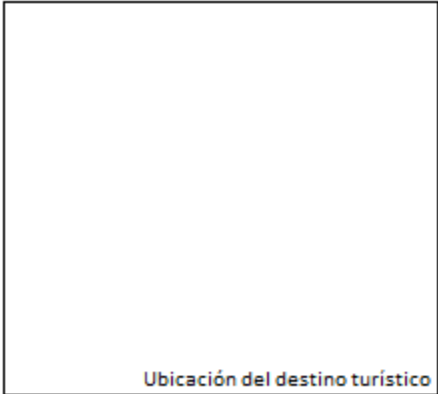
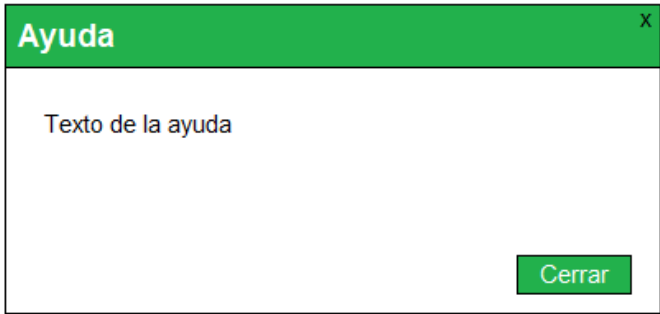
Servicios	Precio	Ubicación
	CUP por noche	 <p>Ubicación del destino turístico</p>
	Contacto	
	Teléfono(s):	
	Correo:	

Tabla 10: Historia de usuario #9

Historia de Usuario	
Número: 9	Nombre: Mostrar ayuda
Usuario: Administrador	
Prioridad en Negocio: Baja	Iteración Asignada: 3
Riesgo en Desarrollo: Bajo	Puntos Estimados: 0.2
Descripción: El sistema muestra una ayuda sobre cómo utilizar la aplicación.	
Observaciones: -	
Prototipo de Interfaz:	
	

2.4 Planificación

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 o gerentes. El proyecto comienza recopilando historias de usuarios, las que sustituyen a los tradicionales casos de uso. Una vez obtenidas las historias de usuarios, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene riesgos que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba, para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala unas pocas historias de usuarios (Joskowicz, 2008).

2.4.1 Estimación del esfuerzo

Para la planificación del desarrollo de la propuesta de solución es necesario estimar de forma ideal el tiempo que tardará el programador en codificar cada una de las HU. Las

condiciones ideales se dan cuando se realiza la codificación sin distracciones y dedicándose a tiempo completo. La estimación de cada HU debe durar un máximo de 3 semanas; en caso de utilizar más tiempo de lo previsto se debe realizar una división de la HU en partes más pequeñas.

Tabla 11: Estimación de esfuerzo

Iteración	Historias de Usuario		Puntos estimados (semanas)
1	1	Autenticar usuario	0.5
	2	Gestionar datos de los destinos turísticos	2.5
2	3	Gestionar datos de los usuarios	2.5
3	4	Buscar destino turístico	0.4
	5	Buscar usuario	0.4
	6	Mostrar cantidad de destinos turísticos	0.5
	7	Mostrar cantidad de usuarios	0.5
	8	Mostrar vista previa del destino turístico	1
	9	Mostrar ayuda	0.2
Total			8.5

2.4.2 Plan de Iteraciones

A partir de la prioridad asignada a cada HU, se decide cuál o cuáles de ellas se implementarán en las primeras iteraciones. Las funcionalidades críticas del sistema deben ser codificadas en iteraciones tempranas.

Iteración 1

Se propone realizar la codificación de las HU “Autenticar usuario” y “Gestionar datos de los destinos turísticos” debido a su alta prioridad para el negocio. En esta iteración se implementarán las funcionalidades principales del sistema que incluyen autenticar usuario para la HU #1 e insertar, modificar, mostrar y eliminar los datos de los destinos turísticos quedando plasmados en la HU #2. De esta forma queda liberada una primera versión del producto o prototipo funcional para el usuario.

Iteración 2

Se propone realizar la codificación de la HU “Gestionar datos de los usuarios” debido a que posee una prioridad media para el negocio. Esta quedará plasmada en la HU #3 para liberar la segunda versión del producto.

Iteración 3

Como última iteración se desarrollarán las HU “Buscar destino turístico”, “Buscar usuario”, “Mostrar cantidad de destinos turísticos”, “Mostrar cantidad de usuarios”, “Mostrar vista previa del destino turístico” y “Mostrar ayuda” debido a que poseen una prioridad baja para el negocio. En esta iteración se implementarán las cinco últimas funcionalidades del sistema, quedando plasmadas en las HU #4, #5, #6, #7, #8 y #9 para liberar la última versión del producto.

Tabla 12: Plan de duración de las iteraciones

Iteración	Historias de Usuario		Duración (semanas)
1	1	Autenticar usuario	3
	2	Gestionar datos de los destinos turísticos	
2	3	Gestionar datos de los usuarios	2.5
3	4	Buscar destino turístico	3
	5	Buscar usuario	
	6	Mostrar cantidad de destinos turísticos	
	7	Mostrar cantidad de usuarios	
	8	Mostrar vista previa del destino turístico	
	9	Mostrar ayuda	
Total			8.5

2.4.3 Plan de entrega

El plan de entrega no es más que la toma de acuerdo del equipo de desarrollo con el cliente sobre el contenido que posee la primera entrega y determinar un intervalo de

tiempo para las restantes entregas del producto. Dicho plan muestra detalles de cómo va a ser monitoreado y controlado incluyendo tolerancias para todos los componentes de entrega relevantes. Mediante el mismo, el equipo de desarrollo mantiene informado y actualizado al cliente de dicho proceso, funcionamiento y la calidad del sistema (Manzanera, 2015).

Tabla 13: Fechas de entrega

Iteración	Fecha de entrega
Iteración 1	2019/03/22
Iteración 2	2019/04/12
Iteración 3	2019/05/03

2.5 Diseño

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes (Joskowicz, 2008):

- **Simplicidad:** Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.
- **Soluciones:** Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba para explorar diferentes soluciones. Estos programas son únicamente para probar o evaluar una solución, y suelen ser desechados luego de su evaluación.

2.5.1 Patrones arquitectónicos

Un patrón arquitectónico se puede considerar como una descripción abstracta estilizada de buena práctica, que se ensayó y puso a prueba en diferentes sistemas y entornos. De este modo, un patrón arquitectónico debe describir una organización de sistema que ha tenido éxito en sistemas previos. Debe incluir información sobre cuándo es y cuándo no es adecuado usar dicho patrón, así como sobre las fortalezas y debilidades del patrón (Sommerville, 2011).

Arquitectura microservicios

La arquitectura de Microservicios es un distintivo sistema de desarrollo de software que ha crecido en popularidad en los últimos años. Por su sencilla escalabilidad, este método de arquitectura se considera especialmente adecuado cuando se tiene que procurar la compatibilidad con un amplio sector de diferentes plataformas (Internet de las cosas, aplicaciones web, aplicaciones para teléfonos celulares, entre otros) o simplemente cuando no se sabe a ciencia cierta hacia qué tipo de dispositivos estará orientando el desarrollo de un software. Cabe destacar varias características comunes en mayor o menor medida (Esaú, 2016):

- El software construido en base a microservicios se podrá descomponer en varias partes funcionales independientes.
- La forma en la que se organizan los microservicios suele ser en torno a las necesidades, capacidades y prioridades del cliente o negocio en el que se implantará.
- Cada módulo contará por ejemplo con su propia base de datos, en lugar de acudir todos a la misma sobrecargándola con solicitudes y con el riesgo de que si falla esta, toda la aplicación pudiera fallar.
- Cuando varios servicios están comunicados entre sí, por lo general contarán con un sistema de aviso y actuación si alguno de estos servicios llega a fallar.

2.5.2 Patrones de diseño

Un patrón de diseño es una descripción de un problema y la solución, a la que se le da un nombre, y que se puede aplicar a nuevos contextos. Proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos. Muchos patrones proporcionan guías sobre el modo en el que debería asignarse las responsabilidades a los objetos, dada una categoría específica del problema (Larman, 2003).

Un patrón de diseño es una forma reutilizable de resolver un problema común (Leiva, 2016).

Patrones GRASP

GRASP es un acrónimo de *General Responsibility Assignment Software Patterns* (patrones generales de software para asignar responsabilidades). El nombre se eligió

para sugerir la importancia de aprender estos principios para diseñar con éxito el software orientado a objetos. Ejemplos de patrones GRASP (Larman, 2003):

- **Experto en información:** se encarga de asignar una responsabilidad al experto en información, o sea, aquella clase que cuenta con la información necesaria para cumplir la responsabilidad. En el módulo informático este patrón se evidencia en la clase `UsuarioService` pues es la única con la responsabilidad de mostrar todos los datos de los usuarios que intervienen en el sistema.
- **Creador:** este patrón es el responsable de asignarle a la clase B la responsabilidad de crear una instancia de la clase A. B es un creador de objetos A. Dicho patrón realiza las funciones de insertar, modificar, mostrar y eliminar datos en la base de datos. En el módulo informático este patrón se evidencia en el servicio `DestinoTuristicoService` que está asociado a la clase entidad existente y se encargan de insertar, modificar, mostrar y eliminar en la base de datos los objetos de dicha clase entidad.
- **Alta cohesión:** asigna una responsabilidad de forma tal que la cohesión siga siendo alta. En el módulo informático este patrón se evidencia en el servicio `DestinoTuristicoService` pues el mismo necesita conocer la ubicación del destino turístico y la responsabilidad de brindar esa información la tiene la clase `UbicacionDTService`, reafirmando la alta cohesión puesto que `DestinoTuristicoService` no implementa este método, solo hace uso de él.
- **Bajo acoplamiento:** asigna una responsabilidad de forma tal que el acoplamiento siga siendo bajo. Este patrón está presente en el módulo informático porque no existe dependencia de una clase a otra, por tanto se mantiene bajo.
- **Controlador:** asigna la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. En el módulo informático este patrón se evidencia ya que, para cada petición o evento generado en el mismo, existe un controlador que es el encargado de obtenerla y devolver una respuesta.

2.5.3 Modelo de datos

El modelado de datos es una manera de estructurar y organizar los datos para que se puedan utilizar fácilmente por las bases de datos (tecnologías-información, 2018).

El modelo de datos que se muestra a continuación en el diagrama 1 fue realizado mediante la herramienta Visual Paradigm. Dicho modelo es necesario para poder

entender de una mejor forma el módulo informático propuesto, que permitirá solucionar las necesidades que tiene de la empresa XETID, que fueron planteadas en la investigación por parte del autor.

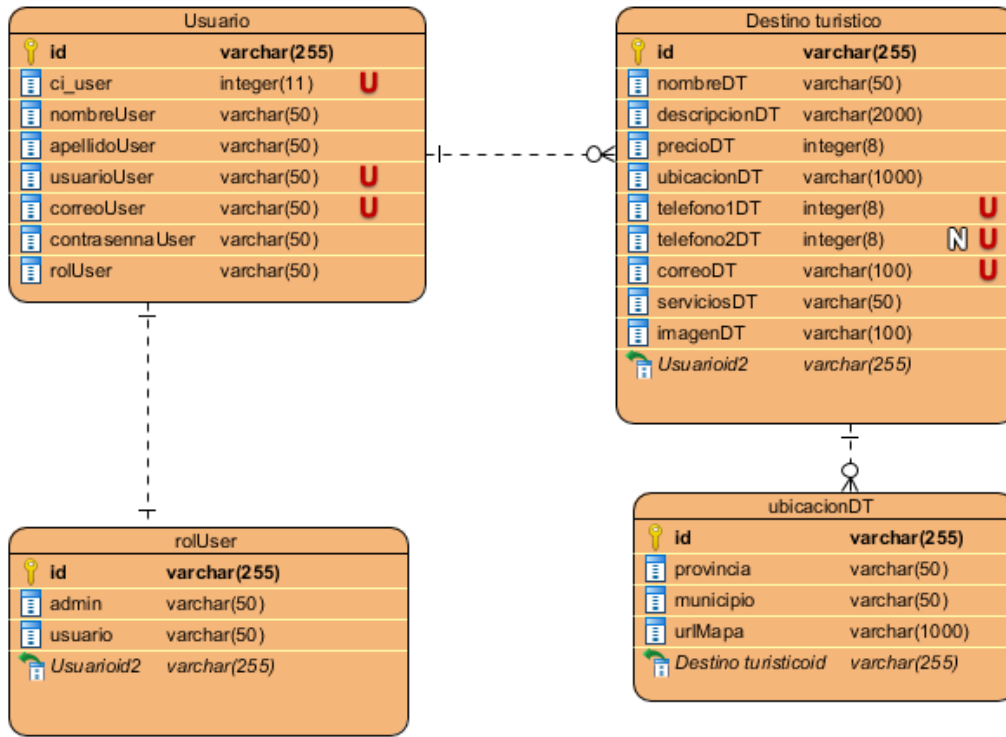


Figura 9: Modelo de datos del módulo informático (Elaboración propia)

2.5.4 Tarjetas Clase-Responsabilidad-Colaboración (CRC)

Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades. En la práctica conviene tener pequeñas tarjetas de cartón, que se llenarán y que son mostradas al cliente, de manera que se pueda llegar a un acuerdo sobre la validez de las abstracciones propuestas. Los pasos a seguir para llenar las tarjetas son los siguientes (Balarezo Penadillo, y otros, 2013):

- Encontrar clases.
- Encontrar responsabilidades.
- Definir colaboradores.
- Disponer las tarjetas.

La técnica de las tarjetas CRC tiene como uno de sus usos guiar el sistema a través de análisis guiados por la responsabilidad. Las clases se examinan, se filtran y se refinan en base a sus responsabilidades con respecto al sistema, y las clases con las que necesitan colaborar para completar sus responsabilidades (Popkin Software and Systems, 1998).

Tabla 14: Tarjeta CRC #1

Tarjeta CRC	
Clase: DestinoTurísticoService	
Responsabilidad	Colaboración
Esta clase es la encargada de gestionar todos los datos del destino turístico que insertan en el sistema el administrador o los usuarios.	DestinoTurístico UbicaciónDTService TipoDTService UsuarioService

Tabla 15: Tarjeta CRC #2

Tarjeta CRC	
Clase: UsuarioController	
Responsabilidad	Colaboración
Esta clase es la encargada de gestionar todos los datos de los usuarios que van a interactuar con el sistema.	UsuarioService rolUserService DestinoTurísticoService

Tabla 16: Tarjeta CRC #3

Tarjeta CRC	
Clase: UbicaciónDTService	
Responsabilidad	Colaboración
Esta clase es la encargada de proporcionar la provincia, el municipio y la ubicación geográfica del destino turístico.	UbicaciónDT DestinoTurístico

Tabla 17: Tarjeta CRC #4

Tarjeta CRC	
Clase: rolUserService	
Responsabilidad	Colaboración

Esta clase es la encargada de asignar el rol correspondiente a los usuarios registrados en el sistema.

rolUser
UsuarioService

2.6 Conclusiones del capítulo

La confección de las 9 HU identificadas en la fase de exploración contribuyó a comprender cuáles eran las necesidades del sistema planteadas por el cliente. El resultado de realizar la estimación por esfuerzo en la fase de planificación, permitió estimar el tiempo que tardaría el programador en realizar la codificación de cada una de las HU identificadas. De igual modo, la selección de la arquitectura y los patrones de diseño, contribuyeron a realizar una implementación centrada en el diseño realizado.

Capítulo 3. Implementación y pruebas

3.1 Introducción

Después de analizadas las 3 primeras fases de la metodología de desarrollo XP propuesta por Kent Beck, se procede a analizar en el capítulo 3 la fase de implementación y pruebas como la última a tratar para darle cumplimiento al alcance de la propuesta de solución de esta investigación. Se exponen además, las tareas de ingeniería generadas por cada HU para realizar la implementación del sistema. Luego se procede a realizarle las pruebas al módulo informático para validar los resultados obtenidos en él.

3.2 Implementación

En esta fase las historias de usuario que quedaron definidas serán implementadas, cumpliendo además con el plan de iteraciones propuesto. Las tareas de implementación estarán basadas en los patrones y estilos arquitectónicos antes propuestos, para de esta forma obtener un software reutilizable. En la fase de implementación el cliente debe estar muy compenetrado con el equipo de desarrollo, ya que el mismo proporciona las historias de usuario y las mismas son expresamente cortas y de alto nivel, además de que no contienen los detalles necesarios para realizar el desarrollo del código (Joskowicz, 2008).

XP propone que la implementación del código sea realizado a través de pares de programadores, ya que al ejecutar esta práctica se minimizan los errores y se logran mejores diseños. Esta buena práctica proporciona que los problemas sean solucionados de forma más rápida, que el proyecto termine con más personas que conozcan los detalles de cada parte del código y que el equipo aprenda a trabajar unido generando una buena dinámica y que la información fluya rápidamente. Otra recomendación importante de esta metodología es la utilización de estándares, para lograr que el código sea entendible y que se faciliten las tareas de recodificación (Joskowicz, 2008).

3.2.1 Tareas de ingeniería

A través de las tareas de ingeniería (TI) los desarrolladores tendrán una guía por la cual orientarse para la implementación del código. Dichas tareas son escritas en

lenguaje técnico, lo que le posibilita al programador una visualización de la estructura a nivel de código del software.

Cuando el equipo de desarrollo conforma las TI deja bien claro en sus campos a qué HU pertenece. Los programadores responsables de implementar dicha funcionalidad, establecen fecha de inicio y de fin en correspondencia con los datos que aporta la HU en cuestión. Los puntos estimados indican el tiempo real que se demoraron los programadores en realizar la funcionalidad. El tipo de tarea que se le va a realizar, el nombre de la misma, y la descripción de la tarea; en este campo como su nombre lo indica se detalla todo lo que se va a realizar y en lo que se va a apoyar para darle solución a la funcionalidad correspondiente.

Tabla 18: Tarea de ingeniería # 1

Tarea	
Número de la tarea: 1	Número de la HU: 1
Nombre de la tarea: Autenticar usuario	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 2019-02-13	Fecha de fin: 2019-02-15
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: Una vez en la página de autenticación, si el administrador introduce su usuario y contraseña y son válidos al presionar el botón “Acceder”, entonces accede a la página de administración. Si fue un usuario el que se autentica correctamente, accede a la página perteneciente a los usuarios. Si no se introduce el usuario y la contraseña, se muestra el mensaje: “Error, introduzca su usuario y contraseña”. Si se introduce un usuario y no se introduce una contraseña, se muestra el mensaje: “Error, introduzca su contraseña”. Si se introducen usuario y contraseña de forma errónea, se muestra el mensaje: “Error, usuario o contraseña incorrecta”.	

Tabla 19: Tarea de ingeniería # 2

Tarea	
Número de la tarea: 2	Número de la HU: 2
Nombre de la tarea: Insertar datos de los destinos turísticos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 2019-02-18	Fecha de fin: 2019-02-21
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: El administrador o los usuarios del sistema, una vez autenticados,	

presionan el botón “Crear destino (+ Nuevo)”, y seguidamente se llenan todos los campos del formulario que se muestra, introduciendo la información correspondiente para crear un destino turístico. Finalmente presiona al botón “Aceptar” para guardar la información en el sistema, o “Cancelar” en caso contrario. Si presiona “Aceptar” se muestra una ventana confirmando que se creó el destino turístico satisfactoriamente. Si presiona “Cancelar” se muestra una ventana para confirmar si realmente desea cancelar o no la operación que esté realizando. Si se autenticó el administrador, puede crear todos los destinos turísticos que sean necesarios. Si se autenticó un usuario, solo puede crear un solo destino turístico. Para completar el formulario que se muestra, tener en cuenta los siguiente:

- Campo ***Nombre**: Solo acepta letras y números, incluyendo el caracter espacio. En caso contrario se mostrará el siguiente mensaje: “Error, solo letras y números”. Si se encuentra vacío se mostrará el siguiente mensaje: “Error, introduzca un nombre”.
- Campo ***Alojamiento**: Permite seleccionar el tipo de alojamiento según el destino turístico. Los tipos de alojamiento son: Hotel, Hostal, Hotel-Apartamento, Motel, Campismo, Villa turística. Si no se selecciona un tipo de alojamiento, se muestra el mensaje: “Seleccione”. Si se selecciona la opción Hotel, debe mostrarse un componente que permita seleccionar la cantidad de estrellas que puede clasificar a un hotel (de 0 a 5 estrellas). En caso de seleccionarse otra opción, este componente debe ocultarse.
- Campo ***Descripción**: Acepta letras y números. Además también acepta los siguientes caracteres especiales: guion bajo, signos de exclamación, dos puntos, punto, coma, signos de interrogación, punto y coma, por ciento, símbolo de peso, número, comillas, paréntesis, guion medio. En caso contrario se mostrará el siguiente mensaje: “Error, ha introducido caracteres no válidos”. En caso de estar vacío se mostrará el siguiente mensaje: “Error, la descripción no puede estar vacía”.
- Campo ***Precio**: Solo acepta números y el caracter especial punto (después del punto van los centavos, ejemplo: 210.45). En caso contrario se mostrará el mensaje: “Error, precio no válido”. En caso de estar vacío se muestra el siguiente mensaje: “Error, introduzca un precio”.
- Campo ***Provincia**: Permite seleccionar la provincia en la que se encuentra el destino turístico. Si no se selecciona la provincia, se muestra el mensaje: “Seleccione”. Si se selecciona una provincia, debe mostrarse un componente que permita seleccionar el municipio deseado en correspondencia con la

provincia seleccionada. Si no se selecciona algún municipio, se muestra el mensaje: “Seleccione”.

- Campo ***Teléfono**: Único para cada destino turístico. Solo acepta una cadena de números de longitud 8. Si la cadena de números introducida tiene longitud distinta de 8, se muestra el siguiente mensaje: “Error, teléfono no válido”. Si se introduce una cadena no numérica, se muestra el siguiente mensaje: “Error, solo números sin espacios”. En caso de estar vacío se muestra el mensaje: “Error, introduzca un teléfono”. Si se introduce un teléfono que ya existe en el sistema asociado a otro destino turístico, se muestra el siguiente mensaje: “Este teléfono ya existe”.
- Campo ***Correo**: Único para cada destino turístico. Tiene que estar en el formato: usuario@dominio.dominio, si no cumple con esto se muestra el siguiente mensaje: “Error, verifique el correo”. Si el campo se encuentra vacío se muestra el siguiente mensaje: “Error, introduzca un correo”. Si se introduce un correo que ya existe en el sistema asociado a otro destino turístico, se muestra el siguiente mensaje: “Este correo ya existe”.
- Campo ***Mapa**: En este campo debe aparecer la ubicación geográfica del destino turístico según la plataforma de Google Maps. Para ello acceda a Google Maps y siga los siguientes pasos:
 1. Busque la ubicación geográfica del destino turístico que va a crear y selecciónela con un click.
 2. En el menú que aparece a la izquierda en la parte superior, presionar el botón que posee tres líneas horizontales.
 3. Seleccionar la opción “Compartir o insertar mapa”.
 4. Seleccionar la opción “Insertar mapa”.
 5. Cambiar de tamaño mediano a pequeño.
 6. Copiar el código (iframe) que se muestra.
 7. Finalmente pegar el código anteriormente copiado en el campo **Mapa**.Si no se introduce el enlace, se muestra el siguiente mensaje: “Error, introduzca el enlace del mapa”. Si se introduce un enlace no válido, se muestra el siguiente mensaje: “Error, enlace no válido”.
- Campo ***Servicio**: Permite crear un listado de los servicios que se brindan en un destino turístico determinado. Para ello se debe teclear el nombre del servicio y presionar el botón añadir (botón verde con un signo de +) para agregar el servicio; seguidamente se muestra el mensaje: “Servicio añadido”. Si se desea eliminar un servicio del listado de servicios, presione el botón

eliminar (botón rojo con una X); seguidamente se muestra el mensaje: “Servicio eliminado”. Si se va a añadir un servicio que ya existe en el listado de servicios se muestra el mensaje: “El servicio ya existe”. Si no se introduce ningún servicio y se presiona el botón añadir, se muestra el mensaje: “Introduzca el servicio”. Si no se introduce ningún servicio al presionar el botón Aceptar, se muestra el mensaje: “Añada un servicio”.

- Campo ***Imagen**: Este campo es para seleccionar la imagen que va a identificar al destino turístico. Solo se admiten imágenes en ciertos formatos (.jpg .jpeg .png .gif .bmp). Si se introduce una imagen con un formato no válido se muestra el mensaje: “Imagen no soportada”. Si no se introduce una imagen se muestra el mensaje: “Seleccione una imagen”.

Tener en cuenta que todos los componentes que posean un asterisco son obligatorios. Si coincide el nombre, el tipo de alojamiento y el municipio del destino turístico que se está creando con el nombre, el tipo de alojamiento y el municipio de otro destino turístico que se encuentre almacenado en el sistema, se debe mostrar el mensaje: “Este destino turístico ya se encuentra almacenado en el sistema”, por tanto se debe evitar almacenar la información en este caso.

Tabla 20: Tarea de ingeniería # 3

Tarea	
Número de la tarea: 3	Número de la HU: 2
Nombre de la tarea: Mostrar datos de los destinos turísticos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha de inicio: 2019-02-22	Fecha de fin: 2019-02-25
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: Una vez autenticado el administrador, podrá visualizar los datos de todos los destinos turísticos almacenados en el sistema. En el caso de un usuario, una vez autenticado, podrá visualizar los datos del destino turístico que ha insertado con anterioridad. La información que se mostrará es la siguiente: nombre, alojamiento, precio, provincia, municipio, responsable.	

Tabla 21: Tarea de ingeniería # 4

Tarea	
Número de la tarea: 4	Número de la HU: 2
Nombre de la tarea: Modificar datos de los destinos turísticos	
Tipo de tarea: Desarrollo	Puntos estimados: 1

Fecha de inicio: 2019-02-26	Fecha de fin: 2019-03-01
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
<p>Descripción: Si el administrador o algún usuario del sistema desean modificar los datos de un destino turístico, deben dirigirse al panel de opciones del listado de destinos turísticos y seleccionar el botón naranja que posee el ícono de un lápiz, seguidamente se muestra un formulario con los datos a modificar del destino turístico. Finalmente presiona al botón “Actualizar” para guardar la información en el sistema, o “Cancelar” si quiere mantener los datos sin cambios. Si presiona “Actualizar” se muestra una ventana confirmando que se actualizaron los datos del destino turístico satisfactoriamente. Si presiona “Cancelar” se muestra una ventana para confirmar si realmente desea cancelar o no la operación que esté realizando. En el caso del administrador, puede modificar los datos de todos los destinos turísticos almacenados en el sistema. Si es un usuario, solo puede modificar los datos de su propio destino turístico. Cuando se muestre el formulario para modificar los datos del destino turístico, se debe mostrar toda la información correspondiente a ese destino turístico en cada uno de los campos del formulario. Tener en cuenta todo lo descrito en la tarea de ingeniería #2.</p>	

Tabla 22: Tarea de ingeniería # 5

Tarea	
Número de la tarea: 5	Número de la HU: 2
Nombre de la tarea: Eliminar datos de los destinos turísticos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha de inicio: 2019-03-04	Fecha de fin: 2019-03-06
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
<p>Descripción: Si el administrador o algún usuario del sistema una vez autenticados desean eliminar los datos de un destino turístico, deben dirigirse al panel de opciones del listado de destinos turísticos y seleccionar el botón rojo que posee el ícono de un cesto de basura, seguidamente se muestra una ventana preguntando si realmente desea eliminar ese destino turístico, en caso de eliminarlo presionar el botón “Aceptar”, de lo contrario presionar el botón “Cancelar”. En el caso del administrador puede eliminar los datos de cualquier destino turístico almacenado en el sistema. Si es un usuario, solo puede eliminar los datos de su propio destino turístico.</p>	

Tabla 23: Tarea de ingeniería # 6

Tarea

Número de la tarea: 6	Número de la HU: 3
Nombre de la tarea: Insertar datos de los usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 2019-03-07	Fecha de fin: 2019-03-12
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
<p>Descripción: El administrador una vez autenticado en el sistema, procede a seleccionar la pestaña “Usuarios”, después presiona el botón “Crear usuario (+ Nuevo)”, seguidamente llena todos los campos del formulario que se muestra. Finalmente presiona al botón “Aceptar” para guardar la información en el sistema, o “Cancelar” en caso contrario. Si presiona “Aceptar” se muestra una ventana confirmando que se creó el usuario satisfactoriamente. Si presiona “Cancelar” se muestra una ventana para confirmar si realmente desea cancelar o no la operación que esté realizando. Tener en cuenta lo siguiente:</p> <ul style="list-style-type: none"> • Campo Nombre: Solo admite letras y el caracter espacio. En caso contrario muestra el mensaje: “Error, solo letras”. Si no se introduce el nombre se muestra el mensaje: “Error, introduzca el nombre”. • Campo Apellidos: Solo admite letras y el caracter espacio. En caso contrario muestra el mensaje: “Error, solo letras”. Si no se introducen los apellidos se muestra el mensaje: “Error, introduzca los apellidos”. • Campo Carnet de identidad: Este campo es para el carnet de identidad (CI) de la persona. Solo admite una cadena de números con longitud 11. En caso contrario se muestra el mensaje: “Error, CI no válido”. Si el CI que se introduce coincide con otro CI que se encuentra almacenado en el sistema, se muestra el mensaje: “Este carnet ya existe”. Si no se introduce un CI se muestra el mensaje: “Error, introduzca el CI”. • Campo Usuario: Este campo es para otorgarle a la persona responsable del destino turístico el usuario que utilizará para autenticarse en el sistema. Si se introduce un usuario y este coincide con uno que se encuentra almacenado en el sistema, se muestra el mensaje: “Este usuario ya existe”. Solo admite específicamente letras en minúscula y sin acentos, en caso contrario se muestra el mensaje: “Solo letras minúsculas sin acentos”. Si no se introduce el usuario se muestra el mensaje: “Error, introduzca el usuario”. • Campo Contraseña: Este campo es para otorgarle a la persona responsable del destino turístico la contraseña que va a permitirle autenticarse en el sistema. La contraseña debe ser fuerte (combinación de letras, números, caracteres especiales, longitud mayor que 8 y uso de mayúsculas). A medida 	

que se va tecleando la contraseña, debe mostrarse un mensaje que indique cuando la contraseña tiene fortaleza débil, aceptable o fuerte. Si no se introduce la contraseña se muestra el mensaje: “Error, introduzca la contraseña”. Si la contraseña es débil, no se puede permitir almacenar la información de ese usuario.

- Campo **Repetir contraseña**: Este campo es para verificar que la persona escribió la contraseña correctamente. Si la contraseña de este campo no coincide con la del campo Contraseña, se muestra el mensaje: “Error, no coincide”, y se debe evitar almacenar la información del usuario.
- Campo **Correo**: Este campo es para el correo electrónico de la persona responsable para contactar con ella en caso de ser necesario. Único para cada persona. Tiene que estar en el formato: usuario@dominio.dominio, si no cumple con esto se muestra el siguiente mensaje: “Error, verifique el correo”. Si el campo se encuentra vacío se muestra el siguiente mensaje: “Error, introduzca un correo”. Si se introduce un correo que ya existe en el sistema asociado a otra persona, se muestra el siguiente mensaje: “Este correo ya existe”.

Tabla 24: Tarea de ingeniería # 7

Tarea	
Número de la tarea: 7	Número de la HU: 3
Nombre de la tarea: Mostrar datos de los usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha de inicio: 2019-03-13	Fecha de fin: 2019-03-14
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: Una vez autenticado el administrador del sistema, podrá visualizar los datos de todos los usuarios que se encuentran registrados en el sistema. Para ello debe dirigirse a la pestaña “Usuarios”. En el panel del listado de usuarios registrados en el sistema debe aparecer lo siguiente: nombre, apellidos, carnet de identidad, usuario, correo En el caso de un usuario una vez autenticado, podrá visualizar solamente sus datos (nombre, apellidos, usuario, correo).	

Tabla 25: Tarea de ingeniería # 8

Tarea	
Número de la tarea: 8	Número de la HU: 3
Nombre de la tarea: Modificar datos de los usuarios	

Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 2019-03-15	Fecha de fin: 2019-03-20
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
<p>Descripción: El administrador una vez autenticado en el sistema, debe dirigirse a la pestaña “Usuarios”, panel de opciones del listado de usuarios y selecciona el botón naranja con el icono de un lápiz, seguidamente se muestra un formulario con los datos a modificar del usuario seleccionado. Finalmente presiona al botón “Actualizar” para guardar la información en el sistema, o “Cancelar” en caso contrario. Si presiona “Actualizar” se muestra una ventana confirmando que se creó el usuario satisfactoriamente. Si presiona “Cancelar” se muestra una ventana para confirmar si realmente desea cancelar o no la operación que esté realizando. Tener en cuenta lo siguiente:</p> <ul style="list-style-type: none"> • La información que se muestra en el campo Usuario no debe permitir que se modifique. • Debe existir una opción que permita modificar la contraseña o mantener la contraseña actual. Si se selecciona la opción “Modificar”, deben habilitarse los campos Contraseña y Repetir contraseña. Si se selecciona la opción “Mantener” (opción por defecto al mostrarse el formulario para modificar los datos del usuario) se deshabilitan los campos Contraseña y Repetir contraseña. <p>Para modificar los datos del usuario, tener en cuenta la descripción de la tarea de ingeniería #6.</p>	

Tabla 26: Tarea de ingeniería # 9

Tarea	
Número de la tarea: 9	Número de la HU: 3
Nombre de la tarea: Eliminar datos de los usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 0.25
Fecha de inicio: 2019-03-21	Fecha de fin: 2019-03-22
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
<p>Descripción: El administrador una vez autenticado en el sistema, debe dirigirse a la pestaña “Usuarios”, panel de opciones del listado de usuarios y seleccionar el botón rojo que posee el ícono de un cesto de basura, seguidamente se muestra una ventana preguntando si realmente desea eliminar ese usuario, en caso de eliminarlo presionar el botón “Aceptar”, en caso contrario presionar el botón “Cancelar”. Al eliminarse un usuario, debe eliminarse al mismo tiempo el destino turístico del cual es</p>	

responsable, en caso de tener alguno creado.

Tabla 27: Tarea de ingeniería # 10

Tarea	
Número de la tarea: 10	Número de la HU: 4
Nombre de la tarea: Buscar destinos turísticos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 2019-03-25	Fecha de fin: 2019-03-26
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: El administrador del sistema una vez autenticado, escribe el nombre del destino turístico que desea buscar en el formulario de búsqueda, seguidamente presiona el botón “Buscar” y el sistema muestra los resultados de la búsqueda. Si presiona el botón buscar sin introducir el nombre del destino turístico, se muestra el mensaje: “Introduzca el destino turístico a buscar”. Si el destino turístico introducido no se encuentra almacenado en el sistema, se muestra el mensaje: “No se encontraron resultados”, en caso contrario se muestra el mensaje: “Se encontraron x cantidad de resultados”.	

Tabla 28: Tarea de ingeniería # 11

Tarea	
Número de la tarea: 11	Número de la HU: 5
Nombre de la tarea: Buscar usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 0.4
Fecha de inicio: 2019-03-26	Fecha de fin: 2019-03-27
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: El administrador del sistema una vez autenticado se dirige a la pestaña “Usuarios” y escribe el usuario de la persona responsable que desea buscar en el formulario de búsqueda, seguidamente presiona el botón “Buscar” y el sistema muestra los resultados de la búsqueda. Si se presiona el botón “Buscar” sin introducir el usuario, se muestra el mensaje: “Introduzca el usuario a buscar”. Si el usuario introducido no se encuentra almacenado en el sistema, se muestra el mensaje: “No se encontraron resultados”, en caso contrario se muestra el mensaje: “Se encontraron x cantidad de resultados”.	

Tabla 29: Tarea de ingeniería # 12

Tarea

Número de la tarea: 12	Número de la HU: 6
Nombre de la tarea: Mostrar cantidad de destinos turísticos	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 2019-03-08	Fecha de fin: 2019-03-30
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: El sistema muestra la cantidad de destinos turísticos almacenados.	

Tabla 30: Tarea de ingeniería # 13

Tarea	
Número de la tarea: 13	Número de la HU: 7
Nombre de la tarea: Mostrar cantidad de usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 2019-03-31	Fecha de fin: 2019-04-02
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: El sistema muestra la cantidad de usuarios almacenados.	

Tabla 31: Tarea de ingeniería # 14

Tarea	
Número de la tarea: 14	Número de la HU: 8
Nombre de la tarea: Mostrar vista previa del destino turístico	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 2019-04-02	Fecha de fin: 2019-04-08
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: El administrador o algún usuario, una vez autenticados en el sistema, se dirigen hacia el panel de opciones del listado de destinos turísticos del sistema, y selecciona el botón verde con el ícono que contiene la letra "i" para obtener una vista previa de los datos de un destino turístico específico. En el caso del administrador, podrá obtener la vista previa de los datos de cualquier destino turístico que se encuentre almacenado en el sistema. En el caso de un usuario, solo podrá obtener la vista previa de los datos de su propio destino turístico. La vista previa del destino turístico debe poseer los siguientes datos: nombre, tipo de alojamiento (si es Hotel mostrar la cantidad de estrellas), imagen identificativa, descripción, servicios, precio, teléfono, correo, mapa. La vista previa debe ocultarse si se presiona el botón "Aceptar" o el botón cerrar (ubicado en la parte superior a la derecha).	

Tabla 32: Tarea de ingeniería # 15

Tarea	
Número de la tarea: 15	Número de la HU: 9
Nombre de la tarea: Mostrar ayuda	
Tipo de tarea: Desarrollo	Puntos estimados: 0.2
Fecha de inicio: 2019-04-09	Fecha de fin: 2019-04-10
Programador responsable: Luis Rodolfo Rodríguez Betancourt	
Descripción: El administrador o algún usuario, una vez autenticados en el sistema, seleccionan el botón “Ayuda” y se muestra una ayuda que explica el funcionamiento del sistema. Si se autenticó un usuario, solo se muestra la ayuda con las cosas que solo puede realizar un usuario. Si se autenticó el administrador se muestra la ayuda con las cosas que puede realizar un administrador.	

3.2.2 Pruebas de software

Las pruebas de software intentan demostrar que un programa hace lo que se intenta que haga, así como descubrir defectos en el programa antes de usarlo. Al probar el software, se ejecuta un programa con datos artificiales. Hay que verificar los resultados de la prueba que se opera para buscar errores, anomalías o información de atributos no funcionales del programa. El proceso de prueba tiene dos metas distintas (Sommerville, 2011):

- Demostrar al desarrollador y al cliente que el software cumple con los requerimientos. Para el software personalizado, esto significa que en el documento de requerimientos debe haber, por lo menos, una prueba por cada requerimiento. Para los productos de software genérico, esto quiere decir que tiene que haber pruebas para todas las características del sistema, junto con combinaciones de dichas características que se incorporarán en la liberación del producto.
- Encontrar situaciones donde el comportamiento del software sea incorrecto, indeseable o no esté de acuerdo con su especificación. Tales situaciones son consecuencia de defectos del software. La prueba de defectos tiene la finalidad de erradicar el comportamiento indeseable del sistema, como caídas del sistema, interacciones indeseadas con otros sistemas, cálculos incorrectos y corrupción de datos.

Utilizando la metodología XP, seleccionada para la realización de la investigación en cuestión, se decide la ejecución en el módulo informático de las pruebas unitarias y las

de aceptación. Las pruebas unitarias serán establecidas antes de que se codifique el sistema y serán ejecutadas cada vez que se realice una modificación en el código. Se le aplicarán además las pruebas de aceptación, en la cual el cliente es partícipe constantemente del trabajo y es quien aprueba si se están efectuando correctamente los requisitos funcionales establecidos.

3.2.3 Pruebas unitarias

Una prueba unitaria es la verificación de un módulo (unidad de código) determinado dentro de un sistema. El concepto de “módulo” varía de acuerdo al lenguaje de programación que se esté utilizando. Las pruebas unitarias aseguran que un determinado módulo cumpla con un comportamiento esperado en forma aislada antes de ser integrado al sistema. Los programadores realizan estas pruebas cuando: la interfaz de un método no es clara, la implementación es complicada, para testear entradas y condiciones inusuales, luego de modificar algo. Estas deben contemplar cada módulo del sistema que pueda generar fallas. Para poder integrar el código realizado al ya existente, el mismo debe ser aprobado satisfactoriamente. Las pruebas unitarias permiten la aplicación de otras prácticas como refactorización y diseño simple (Malfará, 2006).

Las pruebas unitarias fueron ejecutadas mediante el marco de prueba de JavaScript: Mocha utilizando la librería Chai. Dichas pruebas fueron realizadas a pequeños pedazos de códigos de las clases controladoras DestinoTurístico y Usuario para garantizar el correcto funcionamiento del código antes de ser programado. A continuación las capturas de pantalla del resultado de las pruebas realizadas:

```

> a@0.0.0 test C:\Users\Luis\Desktop\APIS XETID
> mocha

#test()
  function Checkemail()
    ✓ ok
  function Checkphone()
    ✓ ok
  function Checkprice()
    ✓ ok
  function Checkmaps()
    ✓ ok
  function Checkserv()
    ✓ ok
  function Checkci()
    ✓ ok
  function Checkpw()
    ✓ ok
  function Checkimg()
    ✓ ok
  function Checkname()
    ✓ ok
  function Checkapell()
    ✓ ok

10 passing (46ms)

```

Figura 10: Resultado de la ejecución de las pruebas unitarias

A la propuesta de solución planteada se le realizaron pruebas unitarias a 10 componentes las cuales resultaron 100% efectivas, por lo que el código probado puede ser integrado al módulo informático propuesto.

3.2.4 Pruebas de aceptación

En las pruebas de aceptación el cliente prueba el software y verifica que cumpla con sus expectativas. Estas pruebas generalmente son funcionales y se basan en los requisitos definidos por el cliente y deben hacerse antes de la salida a producción. Las pruebas de aceptación son fundamentales por lo cual deben incluirse obligatoriamente en el plan de pruebas de software. Estas pruebas se realizan una vez que ya se ha probado que cada módulo funciona bien por separado, que el software realice las funciones esperadas y que todos los módulos se integran correctamente.

Tabla 33: Caso de prueba de aceptación # 1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Autenticar usuario	
Descripción: Prueba para la funcionalidad autenticar usuario	
Condiciones de ejecución: El usuario que desee autenticarse debe estar registrado en las BD del sistema.	
Pasos de ejecución: La persona que se autentica debe introducir su usuario y	

contraseña en el formulario de la página de autenticación.

Resultados esperados: El usuario accede correctamente al sistema.

Tabla 34: Caso de prueba de aceptación # 2

Caso de prueba de aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Gestionar datos de los destinos turísticos	
Descripción: Prueba para la funcionalidad gestionar datos de los destinos turísticos	
Condiciones de ejecución: El administrador o algún usuario deben estar autenticados.	
Pasos de ejecución: Luego de autenticado el administrador del sistema, puede acceder a crear, modificar, mostrar y eliminar los datos de todos los destinos turísticos. Si un usuario se autenticó, solo puede crear, modificar, mostrar o eliminar los datos de su propio destino turístico.	
Resultados esperados: Se crean, modifican, muestran o eliminan de forma correcta los datos de los destinos turísticos.	

Tabla 35: Caso de prueba de aceptación # 3

Caso de prueba de aceptación	
Código: HU3_P3	Historia de Usuario: 3
Nombre: Gestionar datos de los usuarios	
Descripción: Prueba para la funcionalidad gestionar datos de los usuarios.	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Pasos de ejecución: Luego de autenticado el administrador del sistema, puede acceder a crear, modificar, mostrar o eliminar los datos de todos los usuarios.	
Resultados esperados: Se crean, modifican, muestran o eliminan de forma correcta los datos de los usuarios.	

Tabla 36: Caso de prueba de aceptación # 4

Caso de prueba de aceptación	
Código: HU4_P4	Historia de Usuario: 4
Nombre: Buscar destino turístico	
Descripción: Prueba para la funcionalidad buscar destino turístico	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Pasos de ejecución: Luego de autenticado el administrador del sistema, puede	

acceder a buscar cualquier destino turístico que se encuentre almacenado en el sistema.

Resultados esperados: Se muestra el resultado de la búsqueda del destino turístico de forma correcta.

Tabla 37: Caso de prueba de aceptación # 5

Caso de prueba de aceptación	
Código: HU5_P5	Historia de Usuario: 5
Nombre: Buscar usuario	
Descripción: Prueba para la funcionalidad buscar usuario	
Condiciones de ejecución: El administrador del sistema debe estar autenticado.	
Pasos de ejecución: Luego de autenticado el administrador del sistema, puede acceder a buscar a la persona responsable de cada destino turístico según el usuario que le fue proporcionado.	
Resultados esperados: Se muestra el resultado de la búsqueda del usuario de la persona responsable de forma correcta.	

Tabla 38: Caso de prueba de aceptación # 6

Caso de prueba de aceptación	
Código: HU6_P6	Historia de Usuario: 6
Nombre: Mostrar cantidad de destinos turísticos	
Descripción: Prueba para la funcionalidad mostrar cantidad de destinos turísticos	
Condiciones de ejecución: Debe existir como mínimo un destino turístico almacenado en el sistema.	
Pasos de ejecución: Una vez que el administrador o algún usuario hayan creado un destino turístico, el sistema mostrará la cantidad de destinos turísticos almacenados.	
Resultados esperados: Se muestra la cantidad de destinos turísticos almacenados según la persona que se autenticó.	

Tabla 39: Caso de prueba de aceptación # 7

Caso de prueba de aceptación	
Código: HU7_P7	Historia de Usuario: 7
Nombre: Mostrar cantidad de usuarios	
Descripción: Prueba para la funcionalidad mostrar cantidad de usuarios	
Condiciones de ejecución: Debe existir como mínimo un usuario almacenado en el sistema.	

Pasos de ejecución: Una vez que se haya creado un usuario (incluyendo al administrador), el sistema mostrará la cantidad de usuarios almacenados.

Resultados esperados: Se muestra la cantidad de usuarios almacenados, solamente siendo visible para el administrador del sistema una vez autenticado.

Tabla 40: Caso de prueba de aceptación # 8

Caso de prueba de aceptación	
Código: HU8_P8	Historia de Usuario: 8
Nombre: Mostrar vista previa del destino turístico	
Descripción: Prueba para la funcionalidad mostrar vista previa del destino turístico	
Condiciones de ejecución: El administrador o algún usuario del sistema deben estar autenticados.	
Pasos de ejecución: Una vez que el administrador o algún usuario hayan creado un destino turístico, se podrá visualizar una vista previa de los datos del destino turístico que fue creado.	
Resultados esperados: Se muestra una vista previa del destino turístico seleccionado de forma correcta, según la persona que se autenticó.	

Tabla 41: Caso de prueba de aceptación # 9

Caso de prueba de aceptación	
Código: HU9_P9	Historia de Usuario: 9
Nombre: Mostrar ayuda	
Descripción: Prueba para la funcionalidad mostrar ayuda	
Condiciones de ejecución: El administrador o algún usuario del sistema deben estar autenticados.	
Pasos de ejecución: Una vez que el administrador o algún usuario se hayan autenticado en el sistema, podrán visualizar una ayuda que les permitirá comprender el funcionamiento del sistema.	
Resultados esperados: Se muestra una ayuda del sistema de forma correcta, según la persona que se autenticó.	

De los 9 casos de prueba de aceptación existió 1 cambio en la primera iteración solicitado por el cliente. Los restantes casos de prueba fueron aceptados por el cliente ya que estuvo de acuerdo con los resultados esperados, quedando aprobada la propuesta de solución planteada luego de corregir el cambio solicitado ([ver anexo 3](#)). A continuación se muestra una figura con los resultados de las pruebas de aceptación.



Figura 11: Resultados de la ejecución de las pruebas de aceptación

3.2.5 Pruebas de integración

Según lo acordado con el cliente, el módulo informático debe integrarse en la plataforma D'Prisa. Por tal motivo deben aplicarse pruebas de integración con el objetivo de verificar que el módulo funcione correctamente en la plataforma, aunque estas pruebas no son propuestas por la metodología de desarrollo de software seleccionada. A continuación se mencionan los pasos que se tuvieron en cuenta para realizar las pruebas de integración:

1. Acceder a la plataforma D'Prisa.
2. Seleccionar el botón que posee el signo más en color verde, que se encuentra al final de la lista de microservicios existentes.
3. Completar el formulario que se muestra en la interfaz con la información necesaria: introducir el nombre del microservicio, definir la información relevante para el funcionamiento y definir las dependencias del microservicio.
4. Ejecutar el microservicio seleccionando el botón con forma de flecha, ubicado a la derecha en la parte superior de la interfaz de la plataforma D'Prisa.

A continuación se muestran las capturas de pantalla relacionadas con el proceso de integración del módulo informático a la plataforma D'Prisa:

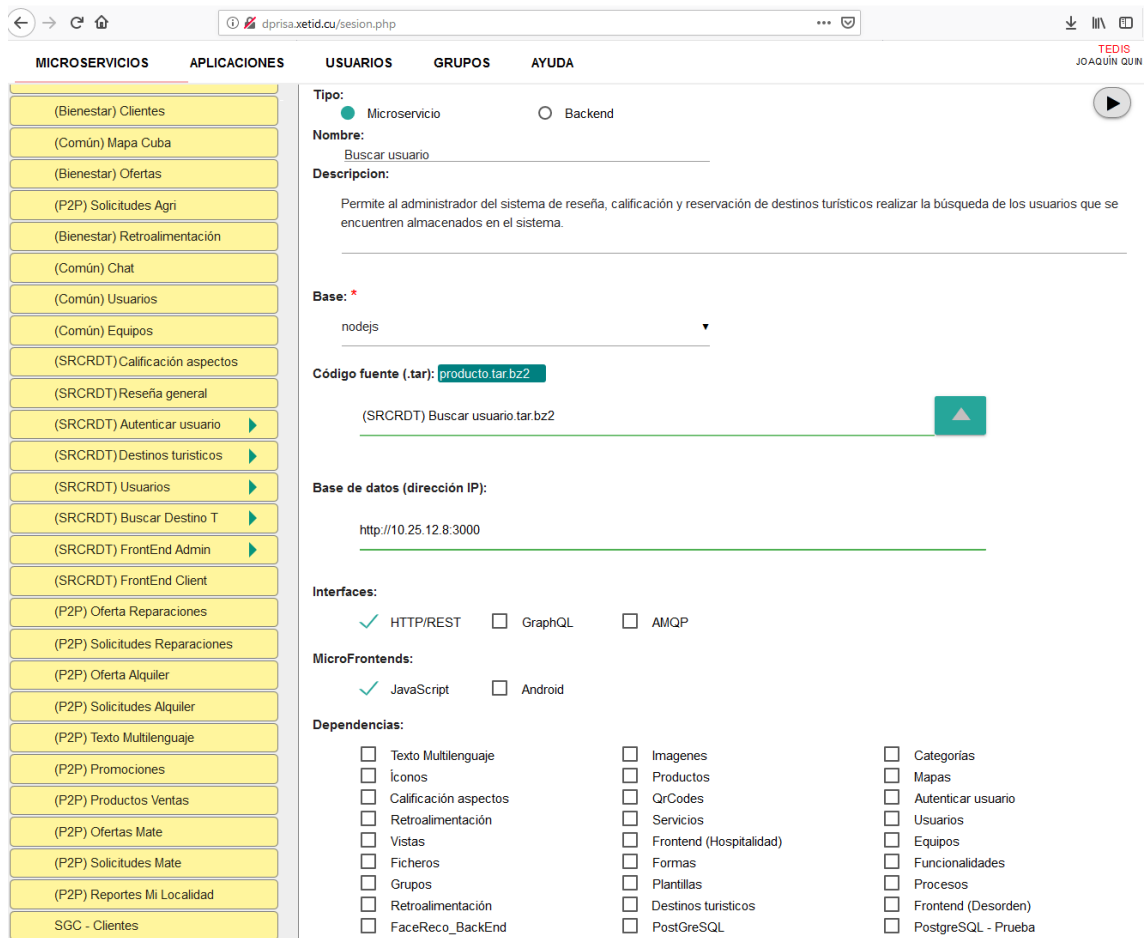


Figura 12: Captura de pantalla de la creación de los microservicios en la plataforma D'Prisa

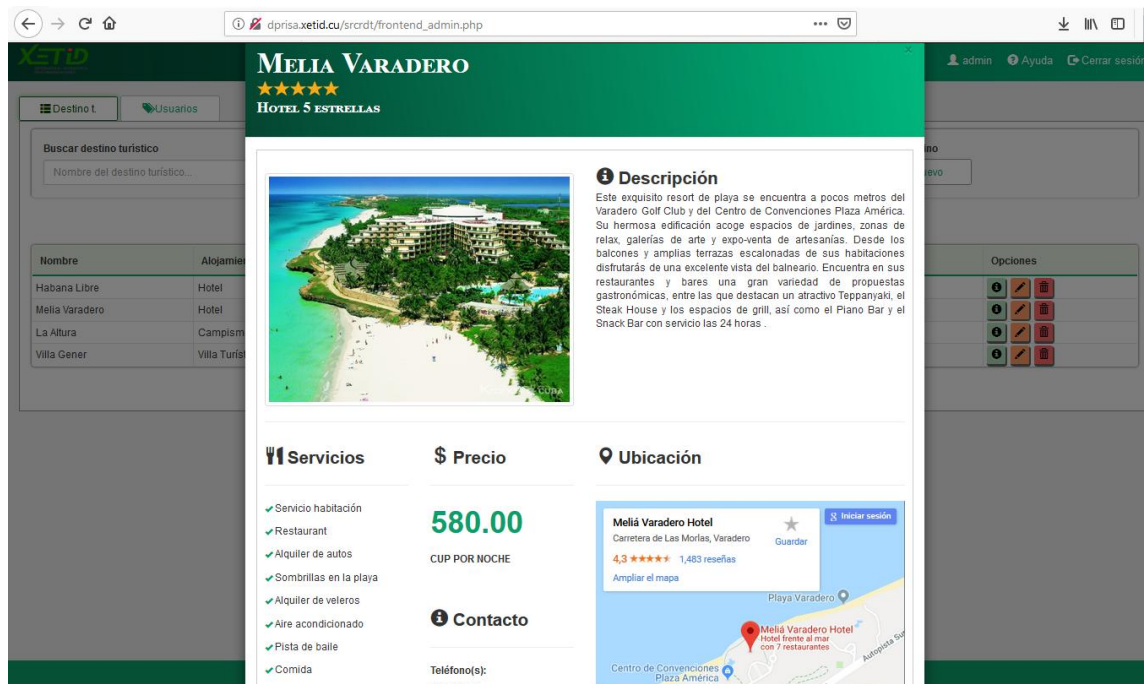


Figura 13: Captura de pantalla del módulo integrado en la plataforma D'Prisa

Luego de realizarse todo este proceso de integración de los microservicios en la plataforma D'Prisa, queda demostrado que el módulo informático en cuestión ha sido integrado correctamente sobre la plataforma. Cumpliendo de esta forma con las expectativas del cliente.

3.2.6 Índice de satisfacción grupal

La técnica ladov se creó inicialmente con el objetivo de obtener el grado de satisfacción de las personas con respecto a la profesión en carreras pedagógicas. Algunos autores la han modificado y aplicado para valorar la satisfacción en múltiples campos y como parte de diagnósticos y validaciones en diferentes investigaciones. Para ello existen tres preguntas fundamentales que se relacionan a través de lo que se denomina el "cuadro lógico de ladov" como se muestra en la siguiente tabla (González Maura, y otros, 2002).

Tabla 42: Cuadro lógico de ladov (Elaboración propia)

	¿Considera usted que se deba reunir la información de todos los destinos turísticos de Cuba en un solo sistema?								
	No			No sé			Si		
	¿Considera útil el módulo implementado para administrar todos los destinos turísticos de Cuba?								
¿Le satisface la propuesta de solución?	Si	No sé	No	Si	No sé	No	Si	No sé	No
Me satisface mucho	1	2	6	2	2	6	6	6	6
No me satisface mucho	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me insatisface más de lo que me satisface	6	3	6	3	4	4	3	4	4
No me satisface nada	6	6	6	6	4	4	6	4	5
No sé qué decir	2	3	6	3	3	3	6	3	4

El uso de la técnica de ladov permite conocer el índice de satisfacción grupal de los usuarios de la XETID. Para obtener sus resultados es necesario conocer la escala de satisfacción, así como la fórmula para determinar el índice de satisfacción grupal.

La escala de satisfacción sería la siguiente:

1. Clara satisfacción
2. Más satisfecho que insatisfecho
3. No definida
4. Más insatisfecho que satisfecho
5. Clara insatisfacción
6. Contradictoria

Luego de haberse triangulado las preguntas del cuadro lógico de ladov, el número resultante de la interrelación de las preguntas indica la posición de cada cual en la escala de satisfacción definida.

Para poder generar el índice de satisfacción grupal se establece una escala numérica entre +1 y -1 como se muestra a continuación:

- +1 Máximo de satisfacción
- +0.5 Más satisfecho que insatisfecho
- 0 No definido
- 0.5 Más insatisfecho que satisfecho
- 1 Máxima insatisfacción

De esta forma ya es posible calcular el índice de satisfacción grupal mediante la siguiente fórmula:

$$\text{ISG} = \frac{A(+1) + B(+0.5) + C(0) + D(-0.5) + E(-1)}{N}$$

El índice de satisfacción grupal, como se especificó en la escala numérica anterior, fluctúa entre + 1 y -1. Es por ello que, una vez calculado, los valores que se encuentren comprendidos entre -1 y -0,5 indican insatisfacción; los comprendidos entre -0,49 y + 0,49 evidencian contradicción y los que se ubiquen entre 0,5 y 1 indican que existe satisfacción.

En la investigación se le aplicó una encuesta de 7 preguntas a una muestra de 15 trabajadores que laboran en la XETID ([ver anexo 2](#)), de ellos 5 administradores de red, 1 jefe de división, 4 programadores y 5 especialistas. La población estuvo compuesta por los trabajadores de la XETID. El valor de índice de satisfacción grupal obtenido fue

de 0.867, encontrándose en el intervalo de satisfacción. De esta forma se puede concluir que el módulo desarrollado posee un alto grado de satisfacción.

3.3 Conclusiones del capítulo

La confección de las 15 tareas de ingeniería que responden a las 9 HU identificadas anteriormente, sirvieron de guía para que el programador pudiera orientarse a la hora de realizar la codificación. De igual modo, la realización de las pruebas al software permitió evaluar la calidad del módulo informático en cuestión, con el objetivo de entregarle al cliente un producto que satisfaga los requerimientos definidos inicialmente.

Conclusiones

Los objetivos propuestos en la investigación han sido cumplidos satisfactoriamente, por tal motivo, se puede concluir que:

- El análisis relacionado con el módulo informático, guiado por la metodología de desarrollo de software XP, permitió obtener una propuesta de solución más detallada a las necesidades que planteaba el cliente de la XETID.
- La implementación del módulo informático contribuyó a que exista una herramienta informática que permita a los usuarios almacenar la información de todos los destinos turísticos de Cuba.
- La ejecución de las pruebas de software validaron que el módulo informático desarrollado sea un producto de calidad, cumpliendo con todas las expectativas del cliente.

Recomendaciones

Las respuestas obtenidas con la investigación proponen algunas recomendaciones que servirían para mejorar el módulo informático propuesto. Por lo que se recomienda:

1. Utilizar el módulo informático en la plataforma D'Prisa.
2. Seguirle realizando pruebas al módulo con el objetivo de encontrarle alguna vulnerabilidad.

Bibliografía

- 1- **Ajax. 2018.** MDN. *MDN*. [En línea] 2018. [Citado el: 30 de noviembre de 2018.] <https://developer.mozilla.org/es/docs/Web/Guide/AJAX>.
- 2- **Alegsa, Leandro. 2018.** Alegsa. *Alegsa*. [En línea] 2018. [Citado el: 14 de noviembre de 2018.] http://www.alegsa.com.ar/Dic/sistema_informatico.php.
- 3- **Apache. 2018.** Apache. *Apache*. [En línea] 2018. [Citado el: 25 de noviembre de 2018.] <http://www.apache.org>.
- 4- **Atrápalo. 2018.** Atrápalo. *Atrápalo*. [En línea] 2018. [Citado el: 14 de noviembre de 2018.] <https://www.atrapalo.com/hoteles>.
- 5- **Aukera. 2019.** Aukera. *Aukera*. [En línea] 2019. [Citado el: 15 de enero de 2019.] <https://aukera.es/blog/bases-de-datos-relacionales-vs-no-relacionales/>.
- 6- **Balarezo Penadillo, Joel Ramos, Cruz Vasquez, Eveling Giselle y Lamadrid Bringas, Franshesca. 2013.** *Metodologías ágiles. Programación extrema XP*. Guadalupe : s.n., 2013.
- 7- **Beck, Kent. 1999.** *Extreme programming explained: embarce change*. Boston : s.n., 1999. ISBN:0-201-61641-6.
- 8- **Booking.com. 2018.** booking.com. *booking.com*. [En línea] 2018. [Citado el: 14 de noviembre de 2018.] <https://www.booking.com>.
- 9- **Bootstrap. 2018.** Bootstrap. *Bootstrap*. [En línea] 2018. [Citado el: 17 de noviembre de 2018.] <http://getbootstrap.com>.
- 10- **Cid, Alejandro Castrelo. 2014.** *MMO de Navegador en Tiempo Real con Node.js y WebSockets*. Barcelona : s.n., 2014.
- 11- **Coello González, Sayda y Hernández León, Rolando Alfredo. 2012.** *EL PROCESO DE INVESTIGACIÓN CIENTÍFICA*. La Habana : Editorial Universitaria, 2012.
- 12- **cubanacan. 2018.** Cubanacan. *Cubanacan*. [En línea] 2018. [Citado el: 15 de noviembre de 2018.] <https://www.hotelescubanacan.com>.
- 13- **Cubazul. 2015.** Cubazul. *Cubazul*. [En línea] 2015. [Citado el: 18 de noviembre de 2018.] <http://cubazul.net/es/cadenas-hoteleras>.
- 14- **Esauí. 2016.** Openwebinars. *Openwebinars*. [En línea] 19 de abril de 2016. [Citado el: 15 de enero de 2019.] <https://openwebinars.net/blog/microservicios-que-son/>.
- 15- **Escribano, Gerardo Fernández. 2002.** *Introducción a Extreme Programming*. 2002.
- 16- **Estándares de Diseño Web. Hernández, Claro y Greugas Navarro, D. 2010.** 2, Ciudad de México : Redalyc, 2010, Vol. Vol. 41.

- 17-Figueroa, Roberth, Solís, Camilo y Cabrera, Armando. 2008.** *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES.* Loja : s.n., 2008.
- 18-Galindo, Luis Hueso Ibáñez. 2011.** *Administración de sistemas gestores de bases de datos.* s.l. : Ra-ma, 2011.
- 19-García, Austin Mata. 2010.** *Un enfoque global de la gestión de los destinos turísticos.* 2010. ISBN:094-5193.
- 20-gaviota. 2018.** Gaviota. *Gaviota.* [En línea] 2018. [Citado el: 15 de noviembre de 2018.] <https://www.gaviota-grupo.com/es/hoteles>.
- 21-González Maura, Viviana y Rodríguez, Alejandro López. 2002.** *efdeportes.* [En línea] 2002. [Citado el: 2019 de abril de 2.] <https://www.efdeportes.com/efd47/iadov.htm>.
- 22-gran-caribe. 2018.** Gran Caribe. *Gran Caribe.* [En línea] 2018. [Citado el: 15 de noviembre de 2018.] <http://www.gran-caribe.cu>.
- 23-Gutiérrez, Yenisel Tirado y Quevedo Rodríguez, Juan Pablo. 2015.** *Plataforma Web Interactiva para el Sistema de Transmisión de Canales Virtuales.* La Habana : s.n., 2015.
- 24-Herramientas CASE para la ingeniería de requisitos. Alarcon, Andrea y Sandoval, Erica. 2008.** 6, s.l. : Cultura Científica, 2008.
- 25-Hotel.com. 2018.** hotel.com. *hotel.com.* [En línea] 2018. [Citado el: 14 de noviembre de 2018.] <https://es.hoteles.com>.
- 26-islazul. 2018.** Islazul. *Islazul.* [En línea] 2018. [Citado el: 15 de noviembre de 2018.] <http://islazul.cu>.
- 27-Joskowicz, José. 2008.** *Reglas y Prácticas en eXtreme Programming.* 2008.
- 28-jQuery. 2018.** jQuery. *jQuery.* [En línea] 2018. [Citado el: 29 de noviembre de 2018.] <https://api.jquery.com/>.
- 29-Kayak. 2018.** Kayak. *Kayak.* [En línea] 2018. [Citado el: 14 de noviembre de 2018.] <https://www.kayak.es/hotels>.
- 30-Kiessling, Manuel y Junge, Herman A. 2015.** *El Libro para Principiantes en Node.js.* 2015.
- 31-Larman, Craig. 2003.** *UML Y PATRONES Una introducción al análisis y diseño orientado a objetos y al proceso unificado.* Madrid : Pearson, 2003. IBN: 84-205-3438-2.
- 32-Leiva, Antonio. 2016.** *Patrones de diseño de software.* 2016.
- 33-Lewis, James y Fowler, Martin. 2014.** MartinFowler.com. *MartinFowler.com.* [En línea] 25 de marzo de 2014. [Citado el: 14 de noviembre de 2018.] <https://martinfowler.com/articles/microservices.html>.

- 34- Maida, Esteban Gabriel y Pacienza, Julián. 2015. METODOLOGIAS DE DESARROLLO DE SOFTWARE.** Buenos Aires : s.n., 2015.
- 35- Malfará, Dayvis. 2006. Testing en eXtreme Programming.** 2006.
- 36- Manzanera, Ignacio. 2015. Plan de entrega.** 2015.
- 37- Metodologías Ágiles para el desarrollo de software: eXtreme Programming (XP). Letelier, Patricio y Penadés, Maria Carmen. 2006.** 26, Buenos Aires : s.n., 2006, Vol. V. ISSN: 1666-1680.
- 38- —. Letelier, Patricio y Penadés, Maria Carmen. 2006.** 26, 2006, Vol. V. ISSN: 1666-1680.
- 39- MINCOM. 2018.** Ministerio de las Comunicaciones de la Republica de Cuba. *Ministerio de las Comunicaciones de la Republica de Cuba.* [En línea] 16 de noviembre de 2018. [Citado el: 18 de noviembre de 2018.] <http://www.mincom.gob.cu/node/1434>.
- 40- Mocha. 2019.** Mocha. *Mocha.* [En línea] 2019. [Citado el: 10 de marzo de 2019.] <https://mochajs.org>.
- 41- MongoDB, Inc. 2008.** MongoDB. *MongoDB.* [En línea] 2008. [Citado el: 14 de enero de 2019.] https://docs.mongodb.com/manual/reference/command/?fbclid=IwAR1EK5dWlZvFEFlg5Mekfez31k4val2IR6NAVBvrg9kpPxes_HnYdeHYvy8.
- 42- muylinux. 2015.** muylinux. *muylinux.* [En línea] 30 de abril de 2015. [Citado el: 4 de diciembre de 2018.] <https://www.muylinux.com/2015/04/30/visual-studio-code-editor-codigo-microsoft-windows-os-x-gnu-linux/>.
- 43- Nadareishvili, Irakli, y otros. 2016. Microservice Architecture.** Sebastopol : O'Reilly, 2016. ISBN: 978-1-491-95625-0.
- 44- Newman, Sam. 2015. Building Microservices.** Sebastopol : O'Reilly, 2015.
- 45- npmjs. 2018.** npm. *npm.* [En línea] octubre de 2018. [Citado el: 20 de marzo de 2019.] <https://www.npmjs.com/package/chai>.
- 46- Popkin Software and Systems. 1998.** Modelado de Sistemas con UML. *Modelado de Sistemas con UML.* [En línea] 1998. [Citado el: 15 de enero de 2019.] <https://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-modelado-sistemas-UML/multiple-html/x98.html>.
- 47- Quintas, Joaquín. 2018. DPrisa.** La Habana : s.n., 2018.
- 48- Real Academia Española. 2018.** Diccionario de la lengua española. *Diccionario de la lengua española.* [En línea] 2018. [Citado el: 6 de diciembre de 2018.] <https://dle.rae.es/?id=1vk04sl>.

- 49- Rodríguez, Liliana Pérez. 2014.** Base de datos. *Base de datos*. [En línea] 21 de agosto de 2014. [Citado el: 15 de enero de 2019.] <http://perezlili.blogspot.com/2014/08/base-de-datos.html>.
- 50- Slideshare. 2014.** Slideshare. *Slideshare*. [En línea] 14 de agosto de 2014. [Citado el: 29 de noviembre de 2018.] <https://es.slideshare.net/rafq007/diferencias-entre-base-de-datos-relacional-y-no-relacional>.
- 51- Sommerville, Ian. 2011.** *Ingeniería de software*. México : Pearson Educación, 2011. ISBN: 978-607-32-0603-7.
- 52- tecnologías-información. 2018.** *Modelado de Datos*. 2018.
- 53- Travelholic. 2017.** Travelholic. *Travelholic*. [En línea] 19 de agosto de 2017. [Citado el: 1 de diciembre de 2018.] <https://www.travelholicperu.com/los-20-mejores-buscadore-hospedaje>.
- 54- tripadvisor. 2018.** TripAdvisor. *TripAdvisor*. [En línea] 2018. [Citado el: 14 de noviembre de 2018.] <https://www.tripadvisor.es>.
- 55- trivago. 2018.** trivago. *trivago*. [En línea] 2018. [Citado el: 14 de noviembre de 2018.] <https://www.trivago.es>.
- 56- tutorialspoint. 2019.** tutorialspoint. *tutorialspoint*. [En línea] 2019. [Citado el: 15 de enero de 2019.] https://www.tutorialspoint.com/mongodb/mongodb_overview.htm.
- 57- UCI. 2018.** UCI. *UCI*. [En línea] 2018. [Citado el: 12 de noviembre de 2018.] <http://www.uci.cu/universidad/mision>.
- 58- Valdés, Damián Pérez. 2007.** Maestros del web. *Maestros del web*. [En línea] 3 de julio de 2007. [Citado el: 16 de noviembre de 2018.] <http://www.maestrosdelweb.com/que-es-javascript/>.
- 59- visual-paradigm. 2018.** visual-paradigm. *visual-paradigm*. [En línea] 2018. [Citado el: 17 de noviembre de 2018.] <https://www.visual-paradigm.com>.
- 60- W3Schools.com. 2019.** AJAX Introduction. *AJAX Introduction*. [En línea] 2019. [Citado el: 24 de febrero de 2019.] https://www.w3schools.com/xml/ajax_intro.asp?fbclid=IwAR2AKsyWrcn2tXgYR6KNkkvSxEDXDivN5QLK4O6eubGO7eRyC-E13Qntrrc.
- 61- —. 2019.** Bootstrap 3 Tutorial. *Bootstrap 3 Tutorial*. [En línea] 2019. [Citado el: 14 de enero de 2019.] <https://www.w3schools.com/bootstrap/?fbclid=IwAR2oK4gRbvpFvgg2RonG6HG5Z9eUBJ774fgc12ykbh2pE3qPtcLWs3aNsek>.
- 62- —. 2018.** jQuery Introduction. *jQuery Introduction*. [En línea] 2018. [Citado el: 29 de noviembre de 2018.] https://www.w3schools.com/jquery/jquery_intro.asp.

- 63- —. 2019.** Node.js MongoDB Create Database. *Node.js MongoDB Create Database*. [En línea] 2019. [Citado el: 23 de febrero de 2019.] https://www.w3schools.com/nodejs/nodejs_mongodb_create_db.asp?fbclid=IwAR25zZVvNZI-0gP3uuQOA1sBpy_t2mplu8mqb3X-mxTHRoi8yO-d9fhqh1w.
- 64- —. 2019.** Node.js Tutorial. *Node.js Tutorial*. [En línea] 2019. [Citado el: 22 de febrero de 2019.] https://www.w3schools.com/nodejs/?fbclid=IwAR1TOq_jDU5cyGeV7hVfm6b6FU2rAoLTAJS9q7SDD-s8lasgWnaTXKa_o-k.
- 65- XETID. 2018.** XETID. *XETID*. [En línea] 2018. [Citado el: 20 de noviembre de 2018.] <http://www.xetid.cu>.

Anexos

Anexo 1. Entrevista

Preguntas de la entrevista al MSc. Joaquín Quintas Santiago de la XETID:

- ¿Actualmente existe algún sistema informático en Cuba que permita buscar destinos turísticos?
- ¿Qué es la plataforma D´Prisa?
- ¿Cómo funciona la plataforma D´Prisa?
- ¿Qué son los microservicios?
- ¿Cómo funcionan y que ventajas trae la implementación de los microservicios?
- ¿Por qué se sugiere la arquitectura de microservicios para la implementación de la aplicación?
- ¿Qué aspectos sugiere para el diseño visual del módulo informático?
- ¿Qué tecnologías y lenguajes de programación web propone para el desarrollo del módulo informático?
- ¿Qué información debe contener el módulo informático para almacenar la información de los destinos turísticos?

Anexo 2. Encuesta de satisfacción de usuarios potenciales

Tabla 43: Encuesta aplicada a los trabajadores de la XETID

Evaluación de la propuesta de solución del módulo destinos turísticos del Sistema de reseña, calificación y reservación de destinos turísticos		
No	Preguntas	Respuestas
1	¿Conoce usted lo que es un sistema de almacenamiento de destinos turísticos?	Si ___ No ___
2	¿Sabe usted los beneficios traería para el país el uso de este tipo de sistemas?	Si ___ No ___
3	¿Alguna vez ha utilizado algún otro sistema que almacene información sobre los destinos turísticos de Cuba?	Si ___ No ___
4	¿Considera usted que se deba reunir la información de todos los destinos turísticos de Cuba en un solo sistema?	Si ___ No ___ No sé ___

5	¿Considera útil el módulo implementado para administrar todos los destinos turísticos de Cuba?	Si ___ No ___ No sé ___
6	¿Cómo valora la interfaz visual de módulo implementado?	Buena ___ Regular ___ Mala ___
7	¿Le satisface la propuesta de solución?	Me satisface mucho ___ No me satisface mucho ___ Me da lo mismo ___ Me insatisface más de lo que me satisface ___ No me satisface nada ___ No sé qué decir ___

Anexo 3. Acta de aceptación

La imagen a continuación evidencia como fue aceptada por parte del cliente la propuesta de solución desarrollada:

UCI Universidad de las Ciencias Informáticas

Acta de aceptación

ACTA DE ACEPTACIÓN

En cumplimiento del Convenio de colaboración con la empresa XETID y en función de la ejecución del proyecto: **Módulo proveedor de destinos turísticos del Sistema de reseña, calificación y reservación de destinos turísticos**, se hace entrega de los productos que se relacionan a continuación:

- Módulo proveedor de destinos turísticos del Sistema de reseña, calificación y reservación de destinos turísticos.

Entrega	Recibe
Nombre y apellidos: Luis Rodolfo Rodríguez	Nombre y apellidos: Joaquín Quintas
Cargo: Estudiante	Cargo: J'Div Tecnologías
Firma: <i>[Handwritten Signature]</i>	Firma: <i>[Handwritten Signature]</i>

Fecha: 15/05/2019

Figura 14: Acta de aceptación