

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA GUÍA
GASTRONÓMICA PARA LA ADMINISTRACIÓN Y UBICACIÓN DE
RESTAURANTES EN ENTORNO WEB**

Tesis para optar por el Título de Ingeniero Informático, que presenta el alumno:

Christian Pérez Ortiz

ASESOR: Ing. Rony Cueva Moscoso

Lima, mayo del 2015

Tabla de Contenido

CAPÍTULO 1: GENERALIDADES	1
1 INTRODUCCIÓN	1
2 PROBLEMÁTICA	2
3 OBJETIVO GENERAL	4
4 OBJETIVOS ESPECÍFICOS	4
5 RESULTADOS ESPERADOS	5
6 MÉTODOS, METODOLOGÍAS Y PROCEDIMIENTOS	6
6.1 RELACIONADOS A LA GESTIÓN DEL PROYECTO	6
6.2 RELACIONADOS A LOS RESULTADOS ESPERADOS	9
7 ALCANCE Y LIMITACIONES	14
8 VIABILIDAD Y JUSTIFICATIVA DEL PROYECTO	15
9 PLAN DE ACTIVIDADES	17
10 MARCO TEÓRICO	21
10.1 CRECIMIENTO DE LA GASTRONOMÍA EN EL PERÚ	21
10.2 COMPETENCIA ENTRE RESTAURANTES	21
10.3 CALIDAD DEL SERVICIO EN RESTAURANTES	22
10.4 PORTALES WEB COMO CANALES DE PUBLICIDAD PARA RESTAURANTES	22
10.5 POSICIONAMIENTO DE PORTALES WEB	23
10.6 SOCIAL MEDIA MARKETING EN RESTAURANTES	24
10.7 FACTORES IMPORTANTES EN LOS COSTOS DE UN RESTAURANTE	24
10.8 SISTEMAS DE INFORMACIÓN: GUÍAS GASTRONÓMICAS	25
10.9 CONSIDERACIONES PARA IMPLEMENTAR TI EN UN RESTAURANTE	26
10.10 MECANISMOS DE GEOLOCALIZACIÓN EN ENTORNOS WEB	27
11 ESTADO DEL ARTE	27
11.1 OBJETIVO DE LA REVISIÓN	27
11.2 CONCLUSIONES SOBRE EL ESTADO DEL ARTE	37
CAPÍTULO 2: ANÁLISIS Y ARQUITECTURA	38
1 IDENTIFICACIÓN DE REQUERIMIENTOS	38
1.1 REQUERIMIENTOS FUNCIONALES	38
1.2 REQUERIMIENTOS NO FUNCIONALES	41
2 ANÁLISIS DE LA SOLUCIÓN	42
2.1 ANÁLISIS TÉCNICO Y ECONÓMICO	42
2.2 CATÁLOGO DE ACTORES	44
2.3 PAQUETES	44
2.4 CASOS DE USO	45
2.5 ESPECIFICACIÓN DE CASOS DE USO	47
2.6 DIAGRAMA DE CLASES	55
3 ARQUITECTURA DE LA SOLUCIÓN	56
3.1 VISTA LÓGICA	57
3.2 VISTA DE IMPLEMENTACIÓN	59
3.3 VISTA DE DESPLIEGUE	62
CAPÍTULO 3: ORDENAMIENTO DE RESTAURANTES	64

1	INTEGRACIÓN CON EL SISTEMA	64
2	PREREQUISITOS DEL ALGORITMO	65
2.1	GEOLocalIZACIÓN EN NAVEGADORES DE INTERNET	65
2.2	FÓRMULA DE HAVERSINE	66
2.3	GOOGLE MAPS API WEB SERVICES	67
3	ALGORITMOS DE ORDENAMIENTO “SHELL”	69

CAPÍTULO 4: ANÁLISIS DE COMENTARIOS **72**

1	INTEGRACIÓN CON EL SISTEMA	72
2	ALGORITMO DE BÚSQUEDA DE CADENAS DE TEXTO “FUERZA BRUTA”	73

CAPÍTULO 5: DISEÑO Y CONSTRUCCIÓN **76**

1	DIAGRAMA DE BASES DE DATOS	76
2	DISEÑO DE INTERFAZ GRÁFICA	78
2.1	INTERFAZ PÚBLICA Y DE CONSUMIDORES	78
2.2	INTERFAZ DE ADMINISTRADORES Y RESTAURANTES	79
3	CONSTRUCCIÓN DEL SISTEMA	83
3.1	HERRAMIENTAS PARA LA IMPLEMENTACIÓN	83
3.2	PATRONES DE PROGRAMACIÓN	85
3.3	ESTÁNDARES DE PROGRAMACIÓN	87
3.4	REUTILIZACIÓN DE CÓDIGO	88
4	PRUEBAS	89
4.1	PLANTEAMIENTO DE CASOS DE PRUEBA	89
4.2	IDENTIFICACIÓN DE LOS CASOS DE PRUEBA	90
4.3	ENTORNO DE PRUEBA	90
4.4	CRITERIOS DE ACEPTACIÓN O RECHAZO	91
4.5	PLANTILLA DE LAS PRUEBAS	91
4.6	RESULTADOS DE LAS PRUEBAS	92

CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES **93**

1	CONCLUSIONES	93
2	RECOMENDACIONES	94

REFERENCIAS BIBLIOGRÁFICAS **96**

Índice de Figuras

Figura 1.1 "Ventas Reales del Sector Restaurantes 2011-2012" (INEI 2012.b)	3
Figura 1.2 "Crecimiento de inversión de capital en TI"	26
Figura 1.3 "Degusta Detalle Restaurante I" (DEGUSTA 2012)	28
Figura 1.4 "Degusta detalle restaurante II" (DEGUSTA 2012)	29
Figura 1.5 "Degusta detalle restaurante III" (DEGUSTA 2012)	29
Figura 1.6 "Degusta Móvil – Restaurante I" (DEGUSTA 2015)	30
Figura 1.7 "Degusta Móvil - Restaurante II" (DEGUSTA 2105)	30
Figura 1.8 "Degusta Móvil - GPS I" (DEGUSTA 2015)	31
Figura 1.9 "Degusta Móvil - GPS II" (DEGUSTA 2015)	31
Figura 1.10 "GPS Restaurante.pe"(GPS RESTAURANTES 2012)	32
Figura 1.11 "Guía Oleo I"(Guía Oleo 2012 S.A.)	33
Figura 1.12 "Guía Oleo II"(Guía Oleo 2012 S.A.)	33
Figura 1.13 "TripAdvisor" (TRIPADVISOR INC. 2012)	34
Figura 1.14 "QUEREMOSCOMER.COM"(QUEREMOSCOMER 2012)	35
Figura 1.15 "Foursquare Búsqueda"	36
Figura 1.16 "Foursquare Restaurante"(FOURSQUARE 2012)	36
Figura 1.17 "Foursquare check-in" (FOURSQUARE 2012)	36
Figura 2.1 "Diagrama de Actores" (Elaboración Propia)	44
Figura 2.2 "Diagrama de Paquete" (Elaboración Propia)	45
Figura 2.3 "Paquete Consumidores" (Elaboración Propia)	46
Figura 2.4 "Paquete Restaurantes" (Elaboración Propia)	46
Figura 2.5 "Paquete Restaurantes" (Elaboración Propia)	47
Figura 2.6 "Diagrama de Clases" (Elaboración Propia)	56
Figura 2.7 "Arquitectura - Vista Lógica" (Elaboración Propia)	58
Figura 2.8 "Diagrama de Componentes" (Elaboración Propia)	60
Figura 2.9 "Diagrama de Despliegue" (Elaboración Propia)	63
Figura 3.1 "Paquete con mecanismo de ordenamiento" (Elaboración Propia)	64
Figura 3.2 "Clases: mecanismo de ordenamiento" (Elaboración Propia)	65
Figura 3.3 "Uso de geolocalización" (Elaboración Propia)	66
Figura 3.4 "Fórmula de Haversine" (GOODWIN 1910)	67
Figura 3.5 "Implementación Fórmula de Haversine" (Elaboración Propia)	67
Figura 3.6 "Consulta a Servicios de Google" (Elaboración Propia)	68
Figura 3.7 "Respuesta de Servicio de Google" (Elaboración Propia)	69
Figura 3.8 "Implementación Algoritmo Shell" (Elaboración Propia)	71
Figura 4.1 "Paquete con mecanismo de análisis de texto" (Elaboración Propia)	72
Figura 4.2 "Clases involucradas con mecanismo de texto" (Elaboración Propia)	73
Figura 4.3 "Implementación Algoritmo Fuerza Bruta" (Elaboración Propia)	75
Figura 5.1 "Diagrama de Base de Datos" (Elaboración Propia)	77
Figura 5.2 "Interfaz Pública" (Elaboración Propia)	79
Figura 5.3 "Consulta Restaurantes" (Elaboración Propia)	79
Figura 5.4 "Detalle Restaurante" (Elaboración Propia)	80
Figura 5.5 "Interfaz Restaurante - Administrador" (Elaboración Propia)	81
Figura 5.6 "Detalle Plato" (Elaboración Propia)	81
Figura 5.7 "Editar Restaurante" (Elaboración Propia)	82
Figura 5.8 "Capas de MVC3" (Elaboración Propia)	83
Figura 5.9 "Motor Gráfico Razor" (Elaboración Propia)	84
Figura 5.10 "Patrón DAO" (Elaboración Propia)	86
Figura 5.11 "Estándar Propiedades" (Elaboración Propia)	87

Figura 5.12 "Estándar Parámetros" (Elaboración Propia)88



CAPÍTULO 1: GENERALIDADES

En este capítulo se presentan los aspectos generales del proyecto. Se empieza describiendo el problema identificado, el cual luego es mejor explicado en el marco teórico. Asimismo, se presentan los objetivos, resultados y metodologías que se utilizarán para la elaboración del proyecto, así como el estado del arte para mostrar las formas en las que se ha ido tratando de resolver el problema en los últimos años.

1 Introducción

En el actual contexto peruano, la gastronomía peruana es un mercado en constante movimiento tanto de alzas como de bajas. Esto quiere decir, que así como se van formando nuevos negocios de restaurantes, también van desapareciendo otros. Asimismo, algunos de estos permiten agilizar su ubicación a través de portales web, de tal forma que facilita esta actividad para los consumidores.

Frente a este contexto, aparece el problema de la dificultad de la ubicación de restaurantes. Por un lado, la distribución de portales web a lo largo de Internet, desemboca en una falta de centralización, lo cual no permite a los consumidores ubicar fácilmente un restaurante que se acomode a sus necesidades. Por otro lado, no todos los restaurantes cuentan con un portal web propio, lo cual genera una falta de medios de comunicación entre los restaurantes y clientes.

En este proyecto de fin de carrera, se brindará una propuesta de solución tanto al problema de descentralización de los restaurantes en Internet como a la falta de medios de comunicación entre restaurantes y consumidores.

Para desarrollar esta solución se abarcará el análisis, diseño e implementación de un sistema de información en base a metodologías y procedimientos de ingeniería de software. Asimismo, se realizará el desarrollo de algoritmos para brindar soporte a algunas funcionalidades del sistema.

2 Problemática

Uno de los negocios más importantes y que mayor crecimiento ha experimentado en los últimos años en el Perú es la gastronomía. Esto se comprueba con su comportamiento a lo largo del 2009: se registró 10 mil 118 licencias otorgadas, lo cual quiere decir 28 licencias por día (INEI 2011). Asimismo, en Julio del 2012, se registró un crecimiento del 9.57% a comparación del año pasado que se llegó solo a 9.05% (INEI 2012). En la figura 1.1, se visualiza que la gastronomía peruana es un mercado en constante movimiento, tanto de crecimiento como de descenso. Esto debido a distintas causas como la gran cantidad de restaurantes que abren y cierran al año. “... de un total de establecimientos que se inauguran mes a mes, casi el 50% se cierra” (La Gestión: 2012) afirma Nicolai Stakeeff, actual presidente del Subcomité de Gastronomía de la Cámara de Comercio de Lima. Asimismo, la cocina peruana es considerada una de las más variadas y ricas del mundo, gracias a su herencia pre incaica, incaica y la inmigración española, africana, chino-cantonesa, japonesa e italiana, de aquí que reúne, mezcla y acriolla una gastronomía ofreciendo una variedad de platos típicos en constante evolución (GASTRONOMÍA PERUANA 2012). Ya en el 2007, se publicó que Perú era el país con mayor número de platos típicos en el mundo con un total de 491 (RODRIGEZ Y VENTURO 2007). Asimismo, a la par con este boom de la gastronomía, han ido apareciendo diversos tipos de comida, los cuales agrupan a la gran variedad de platillos en el Perú, desde la criolla hasta la asiática. Por otro lado, los servicios que se brindan en los restaurantes hoy en día son variados y dan un plus a los restaurantes, como por ejemplo los restaurantes que poseen desayunos, buffet, wi-fi, entre otros. Estos marcan una gran diferencia en cuanto a si cumplen o no las necesidades de un cliente.

Este crecimiento ha hecho que la forma de ubicar un restaurante que se adecúe a la necesidad de un consumidor no sea una tarea sencilla. Es por esto que comenzaron a aparecer medios que ayudasen a ubicarlos con más rapidez. En un principio se hacía uso de medios tradicionales como las guías telefónicas o guías de restaurantes; sin embargo, hoy las personas hacen uso de Internet, redes sociales o sus móviles para tener un acceso rápido y actualizado a la información que desean.



Figura 1.1 "Ventas Reales del Sector Restaurantes 2011-2012" (INEI 2012.b)

En este contexto, se presentan distintos problemas con respecto a los clientes y a las personas interesadas por ir a degustar a un restaurante, ya que las opciones son muy variadas y cambiantes por lo que la elección no es una tarea sencilla. Por un lado, los medios tradicionales de búsqueda como las guías telefónicas o de restaurantes no son útiles como en el pasado debido a la constante aparición y desaparición de restaurantes en el día a día, es decir, por su incapacidad de poseer información actualizada y administrable.

Por otro lado, los medios digitales, como Internet, han dado la posibilidad a cada restaurante de que cree su propio portal web para poder ser ubicados; sin embargo, esto ha generado dificultad para los consumidores en realizar dicha actividad, ya que la información en Internet no está centralizada ni organizada, es decir, no se tiene un catálogo de restaurantes en la cual el consumidor pueda ubicar lo que necesita rápidamente. Además, la información publicada por los restaurantes en sus portales web, suele ser redactada por trabajadores propios de estos negocios, lo cual limita información acerca de la calidad en los servicios que brindan estos negocios, es decir, no se posee comentarios o críticas en base a experiencias reales en estos locales. Asimismo, no todos los restaurantes se encuentran registrados en Internet por medio de un portal web, por lo que esto empeora la situación, ya que se hace casi imposible la ubicación de dichos negocios.

Por último, existen otros medios digitales como las guías gastronómicas, las cuales han logrado crear un espacio centralizado para poder ubicar los restaurantes e inclusive mostrar el feedback de los servicios prestados a través de críticas de parte de los comensales con un debido control para filtrar los críticas constructivas de aquellas otras que contengan mensajes falsos que buscan desprestigiar a un restaurante. Sin embargo, dentro de una guía gastronómica la cantidad de comentarios o críticas registradas durante un día puede ser elevada, por lo que la tarea operativa de analizarlas se vuelve lenta y demora la publicación de las mismas. De este modo surge la necesidad de un mecanismo que apoye al análisis del contenido de los comentarios para agilizar su publicación.

El problema que se tratará, en este proyecto de fin de carrera, será la falta de centralización y de información que se presenta en los medios digitales de ubicación de restaurantes actuales, así como la necesidad de un mecanismo que apoye al análisis del contenido de los comentarios publicados por comensales dentro de estos medios. Es decir, la falta de un mismo sitio en el que se puedan encontrar agrupados los distintos restaurantes del país y la falta de información real por parte del consumidor con respecto a la calidad de los servicios recibidos. De esta forma se espera brindar una herramienta que permita a los clientes de los restaurantes poder encontrar fácilmente un local que se adecúe a sus necesidades.

3 Objetivo general

Analizar, diseñar e implementar una guía gastronómica para la ubicación de restaurantes en entorno web, la cual permita un análisis automatizado de las críticas de los comensales.

4 Objetivos específicos

1. Elaborar una arquitectura adecuada que dé el soporte necesario para que los restaurantes puedan publicar información sobre los servicios que ofrecen en un

sitio público y, a la vez, permita a los comensales aportar retroalimentación sobre estos.

2. Desarrollar un mecanismo que permita mostrar los restaurantes de forma ordenada en base a la distancia con respecto a los usuarios, para que así el comensal tenga un criterio más con el cual decidir qué restaurante se ajusta mejor a sus necesidades.
3. Desarrollar un artefacto que permita automatizar el análisis de los comentarios registrados por parte de los comensales en el sistema para evitar la publicación de aquellos que se consideren ofensivos.
4. Desarrollar el prototipo funcional del sistema de información de tal forma que se posean todas las funcionalidades necesarias para alcanzar la solución que se busca brindar.

5 Resultados esperados

- Resultado 1 para el objetivo 1: especificación de la arquitectura web cliente servidor, la cual será necesaria para la construcción del prototipo.
- Resultado 2 para el objetivo 2: mecanismo de ordenamiento que utilice una adaptación del algoritmo de ordenamiento Shell para aplicarlo sobre los restaurantes consultados, el cual se base en las distancias entre las posiciones de los restaurantes y el actual usuario, es decir sus longitudes y latitudes.
- Resultado 3 para el objetivo 3: artefacto de análisis de texto que utilice una adaptación del algoritmo de búsqueda de cadenas de texto de Fuerza Bruta, el cual tome como entrada palabras que se consideren ofensivas para buscarlas en los comentarios registrados y decidir si deben ser aprobados o no para su publicación.
- Resultado 4 para el objetivo 4: prototipo del sistema de información implementado, el cual será verificado por el asesor para corroborar que esté cumpliendo todas las funcionalidades a detallar en el análisis.

6 Métodos, metodologías y procedimientos

En esta sección del capítulo 1, se presentarán una serie de métodos, metodologías, procedimientos y las herramientas que se utilizarán para elaborar los resultados esperados presentados en la sección anterior. Asimismo, se hará una diferenciación entre aquellas metodologías que guiarán el desarrollo del proyecto, de aquellas que se centrarán en la elaboración del producto.

6.1 Relacionados a la gestión del proyecto

Para la gestión del proyecto a elaborar, se utilizarán ciertos principios del PMBOK, los cuales se mencionarán a continuación.

PMBOK es una norma formal que describe métodos, procesos y prácticas establecidas para profesionales dedicado a la dirección de proyectos. Esta norma menciona grupos de procesos para la dirección de proyectos, entre los cuales se encuentra el Grupo del Proceso de Iniciación, Grupo del Proceso de Planificación, Grupo del Proceso de Ejecución, Grupo del Proceso de Seguimiento y Control, y Grupo del Proceso de Cierre. (PMI 2008)

Para la gestión del proyecto de esta tesis, se utilizarán el Grupo de Procesos de Planificación y Grupo de Procesos de Seguimiento y Control por la importancia de algunos de los procesos que forman parte de estos.

El Grupo de Procesos de Planificación está compuesto por aquellos procesos realizados para establecer el alcance total del esfuerzo, definir los objetivos y desarrollar una línea de acción para alcanzar dichos objetivos. En cuanto a este grupo, se utilizarán los siguientes procesos:

- ✓ Crear la EDT (Estructura de Desglose del Trabajo)

Este proceso consistirá en subdividir los entregables y el trabajo del proyecto en componentes más pequeños y fáciles de dirigir.

Este proceso se usará debido a la ayuda que brinda para poder estructurar el proyecto de tesis a nivel integral, de tal forma que permite elaborar un Gantt y administrar los tiempos e incluso especificar fechas en las que se tenga resultados esperados ya listos.

✓ Definir las Actividades

Este proceso consiste en identificar las acciones específicas a realizar para elaborar los entregables del proyecto.

Este proceso irá acompañado del EDT, ya que brinda mayor especificidad en cuanto a las tareas específicas que se realizarán a lo largo de la elaboración de la tesis.

✓ Secuenciar las Actividades

Este proceso consiste en identificar y documentar las relaciones entre las actividades del proyecto.

Este proceso será el que se utilice a continuación de haber definido las actividades, y será de ayuda gracias a que permitirá visualizar fácilmente la secuencia en la que se deberán realizar las actividades para cumplir con los tiempos establecidos.

✓ Estimar la Duración de las Actividades

Este proceso consiste en establecer aproximadamente la cantidad de períodos de trabajo necesarios para finalizar cada actividad con los recursos estimados.

Este es otro proceso que también brindará ayuda a manejar los tiempos dedicados a las actividades, y así poder controlar el avance del proyecto en su totalidad.

✓ Desarrollar el Cronograma

Este proceso consiste en analizar el orden de las actividades, su duración, los requisitos de recursos y las restricciones del cronograma para crear el cronograma del proyecto.

Con el apoyo de todos los procesos anteriores, este cronograma permitirá realizar un control de los avances del proyecto.

El Grupo de Procesos de Seguimiento y Control está compuesto por aquellos procesos requeridos para supervisar, analizar y regular el progreso del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes. En cuanto a este grupo, se utilizarán los siguientes procesos:

✓ Dar Seguimiento y Controlar el Trabajo del Proyecto

Este proceso consiste en revisar, analizar y regular el avance a fin de cumplir con los objetivos de desempeño definidos en el plan para la dirección del proyecto. Dar Seguimiento implica realizar informes de estado, mediciones del avance y proyecciones. Los informes de desempeño suministran información sobre el desempeño del proyecto en lo relativo al alcance, cronograma, costos, recursos, calidad y riesgos, que puede utilizarse como entrada para otros procesos

Este proceso se llevará a cabo a través de las revisiones semanales que se mantengan con el asesor para ir controlando el avance con respecto a los objetivos definidos. Asimismo, se contará con revisiones parciales por parte del jurado del proyecto.

✓ Controlar el Alcance

Este proceso es el que da seguimiento al estado del alcance del proyecto y del producto, y se gestionan cambios a la línea base del alcance.

Este proceso será muy útil para el proyecto en el caso que aparezca un cambio a lo largo de la elaboración del proyecto. Además establece líneas base que permitirán gestionar cambios con respecto al alcance de verse necesario.

✓ Controlar el Cronograma

Este proceso es el que da seguimiento a la situación del proyecto para actualizar el avance del mismo y gestionar cambios a la línea base del cronograma.

Será esencial para el proyecto, ya que mediante este control se sabrá si se llegará a culminar el proyecto de tesis en el tiempo establecido.

Los Grupos de Procesos que no se utilizarán serán los siguientes:

- ✓ Grupo de Procesos de Iniciación, debido a que este se enfoca en los procesos a realizar para definir un nuevo proyecto, y para este caso, no se necesitará de esta formalidad debido al tamaño del proyecto y a que los involucrados solo son el alumno y los profesores, mas no clientes, entre otros, para lo cual se orienta este Grupo de Procesos.
- ✓ Grupo de Procesos de Ejecución. Si bien es cierto, este involucra procesos que facilitan la dirección del proyecto durante su ejecución; sin embargo, para el caso de este proyecto, bastará con el control del cronograma para una buena ejecución del mismo.
- ✓ Grupo de Procesos de Cierre. Este incluye procesos para finalizar las actividades de todos los grupos de los anteriores procesos. Para el caso de este proyecto de tesis, este grupo no genera mayor valor a la elaboración del proyecto, mas que una formalidad, por lo que no se tomará este principio para el desarrollo de este.

6.2 Relacionados a los resultados esperados

Por otro lado, para la gestión del producto, se utilizarán ciertos principios de distintas metodologías y herramientas que se mapearán con los resultados esperados. A continuación se detalla este punto:

- **Resultado 1: arquitectura web cliente servidor**

Metodología y justificación:

Para el desarrollo de la arquitectura, se ha optado por utilizar el modelo de vistas **4+1 Vistas de Arquitectura** (KRUTCHEN, Philippe), ya que esta permite tener un entendimiento detallado de los componentes que forman parte del sistema y su interacción desde las distintas vistas que este ofrece.

Por otro lado, se utilizará el patrón de diseño **Modelo Vista Controlador (MVC)** descrito por primera vez en 1979 por Trygve Reenskaug. Este consta en separar el

software en tres capas. Una de ella es el modelo, el cual es la representación de la información con la que el sistema operará. Otra capa es la vista, la cual presenta el modelo en un formato adecuado para la interacción con el usuario. La última capa es el controlador, la cual maneja los eventos, producto de las acciones del usuario, invocando al modelo y/o a la vista (REENSKAUG, Trygve 1979).

La razón de utilizar este patrón es que una de las herramientas más útiles en el desarrollo de plataformas web es el uso de marcos de trabajo MVC, ya que permite tener una abstracción entre los elementos que forman parte de la interfaz con la que interactúa el usuario final y los elementos que procesan las acciones del usuario y se conectan con una base de datos. Esto a corto, mediano o largo alcance, permite que el sistema sea más sencillo de mantener y escalar.

Principios a utilizar:

Con respecto al modelo de vistas 4+1 Vistas de Arquitectura, se utilizarán solo tres vistas para la descripción de los componentes del sistema. Por un lado, la vista lógica, ya que permite visualizar las capas abstractas que poseerá el sistema; la vista de implementación, ya que permitirá describir los componentes internos de cada una de las capas lógicas indicadas en la vista anterior; por último, la vista de despliegue, ya que permitirá conocer cómo las vistas anteriores se mapean con el hardware, es decir, los dispositivos físicos que se utilizarán en última instancia.

Con respecto al patrón de diseño MVC, como se mencionó, esta consta de tres capas, este principio será el que guíe el desarrollo de la arquitectura a través de las herramientas que se detallan en la siguiente sección.

Herramientas:

Una de las herramientas principales para el desarrollo de esta arquitectura será el **framework MVC3**, el cual es un framework de .NET para poder desarrollar aplicaciones web siguiendo los patrones de diseño MVC. Asimismo, se hará uso de un **servidor web**, el cual se encargará de alojar el sistema y será al que se le consulte vía **http**.

- **Resultado 2: mecanismo de ordenamiento**

Metodología y justificación:

Para el desarrollo del mecanismo de ordenamiento, será necesaria la adaptación de un algoritmo de ordenamiento. El algoritmo elegido en este caso es el de ordenamiento Shell, el cual consiste en separar la estructura a ordenar en estructuras más pequeñas para recién poder ordenarlas. Es decir, si se hablase de un arreglo, se estaría separando el arreglo en otros más pequeños y así se iría ordenando (Shell 1959).

Este algoritmo es sencillo de implementar como se puede visualizar en su pseudocódigo (el cual se presentará en el capítulo 4 de construcción) y su complejidad de ejecución no supera el $O(N^2)$ (Shell 1959). Asimismo, el ordenamiento se realizará como máximo sobre mil restaurantes, bajo esta condición y la del tiempo de ejecución, el sistema funcionará eficientemente. Por estas razones, para su desarrollo no será necesario realizar una experimentación numérica.

Por otro lado, para poder calcular las distancias entre el usuario y los restaurantes, y así poder ordenarlas, se hará uso de la API de Google Maps para poder hallar la geolocalización de los usuarios, cuya utilización se explicará con más detalle en el capítulo 3.

- **Resultado 3: artefacto de análisis de texto**

Metodología y justificación:

Para el desarrollo del artefacto de análisis de texto, se ha planteando utilizar la adaptación de un algoritmo de búsqueda de cadenas de texto para ubicar palabras ofensivas en los textos de las críticas. El algoritmo elegido en este caso es el de búsqueda de cadenas de texto Fuerza Bruta, el cual consiste en ir comparando el patrón de búsqueda con los distintos segmentos del texto (las palabras por separado). Para esto se utilizará la versión de Fuerza Bruta en la que se compara empezando por la primera letra, la última y la del medio entre el segmento (Start-End-Mid) y el patrón. De este

modo, la complejidad de ejecución será de aproximadamente $O(N^2)$ para que así la eficiencia del sistema no se vea comprometida.

Este algoritmo, similar al algoritmo de ordenamiento, es sencillo de implementar como se podrá verificar en su pseudocódigo, el cual se presentará en el capítulo 4 de construcción.

- **Resultado 4: prototipo del sistema de información**

Metodología y justificación:

Se utilizará la metodología RUP (Rational Unified Process), el cual es un enfoque de desarrollo de software iterativo e incremental, centrado en la arquitectura y manejado por los casos de uso (PER 2003). La metodología RUP tiene la característica de poder ser configurable para grandes o pequeños proyectos. La configuración que se realice permite definir la cantidad de documentación como soporte que se desee. Para el presente proyecto, esto es de gran ayuda, ya que la metodología se puede adaptar al tiempo de este, el cual no es ni muy largo ni muy corto (menor a un año). Por otro lado, sin importar la configuración que se dé, la metodología sigue cuatro fases: concepción, elaboración, construcción y transición. Estas fases permitirán tener un entendimiento claro del sistema por parte de todos los interesados y seguir avanzando con el desarrollo de los artefactos con la predisposición de realizar cambios sobre documentos ya elaborados.

Principios a utilizar:

En la justificación de la metodología se mencionó cuatro fases iterativas. Por un lado, se hará uso de la **fase de concepción**, ya que en esta se enfoca el esfuerzo principalmente en que todos los interesados del proyecto tengan un claro entendimiento del alcance del sistema y los objetivos del mismo, de tal forma que se sepa si se debe continuar o no con el proyecto. Para esto, se debe identificar la problemática que busca solucionar el sistema, los beneficios del software y qué es lo que realizará el producto desde una visión a alto nivel. Esta información normalmente se refleja en un documento de visión; sin embargo, para este proyecto de fin de carrera, se ha considerado que este

primer capítulo tiene la información necesaria para lograr los objetivos de esta primera fase. En cuanto a la decisión de continuar o no con el proyecto, esta se obtendrá finalizando la sustentación del curso de tesis 1. Asimismo, entre los artefactos que se elaborarán en esta fase, se encuentran las primeras versiones de los diagramas de casos de uso y el catálogo de actores, en los cuales aún no se tendrá un detalle completo, pero sí una primera concepción para dar a entender el proyecto desde una perspectiva general.

Por otro lado, se hará uso de la **fase de elaboración**, ya que en esta se centraliza el esfuerzo en especificar a mayor detalle las características del sistema y demostrar que la implementación del sistema es factible y que no correrá riesgos de retrasos. En la primera iteración de esta fase, se buscará cubrir la demostración de la factibilidad del sistema con la implementación de casos de uso claves que influyan en la arquitectura, de tal forma que se pueda demostrar que durante la fase de construcción no aparecerán retrasos inesperados. Asimismo, durante la segunda iteración de esta fase, se habrán culminado, en su mayoría, los artefactos a entregar como el diagrama de casos de uso, el catálogo de actores, la especificación de los casos de uso, el diagrama de clases y el diagrama de base de datos.

Finalmente, se hará uso de la **fase de construcción**, en la cual el esfuerzo se centra en la implementación y prueba de las funcionalidades. No obstante, existe aún la posibilidad de cambios en artefactos ya elaborados como podría ser en la base de datos (estos serían solo cambios pequeños ya que su elaboración a detalle se desarrolló durante la fase anterior). Esta fase culminaría con la presentación de un prototipo estable y lo suficientemente maduro como para cumplir todos los objetivos planteados en la fase de concepción. Para esta fase, se plantea dos iteraciones, una en la cual se tendría un avance parcial del producto final y otra en la cual el sistema abarque todos los requerimientos que se especifican en el capítulo 2.

En cuanto a la fase de transición, no se seguirán sus principios, dado que lo que se busca en este proyecto es realizar un prototipo funcional, el cual se obtiene en la fase de construcción.

Herramientas:

Para la elaboración de los documentos (artefactos) que son parte del análisis como el catálogo de usuarios, diagrama de paquetes, entre otros, se usará la herramienta **StarUML**, ya que está diseñada para brindar apoyo en el desarrollo de dichos documentos. Asimismo, para la elaboración del modelo de base de datos se utilizará el ORM (object-relational mapping) **Entity Framework de Visual Studio 2010** y el cliente **SQL Server 2008 Management Studio**, ya que el Entity Framework permite generar el script de la base de datos a partir de las clases creadas en el proyecto, y el cliente del SQL Server permitirá mostrar el modelo final. Por otro lado, para la implementación del prototipo, se utilizará el **framework MVC3**, por lo que el sistema será implementado en **C#**. Asimismo, para la interacción con la base de datos se utilizará el ORM **Entity Framework** que se mencionó previamente. Finalmente, el motor de la base de datos será **SQL Server 2008** y la tecnología **AJAX** para realizar las consultas hacia este.

7 Alcance y limitaciones

En cuanto al alcance del proyecto, este contemplará la presentación del prototipo funcional de la guía gastronómica mencionada en el objetivo general, el cual como solución, tendrá en cuenta los siguientes aspectos:

- El proyecto está orientado para que restaurantes de todo tamaño puedan disfrutar de sus beneficios (grandes o pequeños).
- El sistema contará con la capacidad de que los restaurantes puedan registrarse en este y tengan un sitio para cada uno, en el cual puedan publicar información sobre ellos mismos.
- En cuanto a la arquitectura, se considerará que como máximo se tendrán 1,000 restaurantes registrados. Esto debido a que el sistema que se implementará será un prototipo y no un producto comercial. Además, en el Perú según se mencionó en la problemática, llega a tener un aproximado de 10 mil nuevos restaurantes al

año. Para que esta cantidad de usuarios pudiesen ser soportados por la arquitectura, se tendría que hacer una inversión mayor en el *hardware* a utilizarse, lo cual no será parte del objetivo.

- En cuanto a los tiempos de ejecución de los algoritmos, estos no serán verificados por medio de una experimentación numérica debido a que esta tesis se enfoca en el análisis, diseño e implementación de una solución y no en el desarrollo de un algoritmo. Estos algoritmos a utilizar son de naturaleza sencilla de tal forma que las referencias serán suficientes como para sustentar que cumplan con el tiempo mencionado en los resultados esperados.
- La solución del proyecto permitirá que los clientes puedan calificar y criticar a los restaurantes para realizar una retroalimentación hacia estos, y así se tenga una información más objetiva sobre los servicios de cada uno.
- El sistema pretende integrar a los consumidores y restaurantes en un mismo espacio para que se agilice la comunicación entre ellos, y así se pueda generar una guía de restaurantes con amplia información.
- El sistema será desarrollado y verificado por medio de un prototipo el cual será implantando en un ambiente de pruebas. No se pretende implantarlo en un ambiente real.

En cuanto a las limitaciones, no se tendrá mayor obstáculo en la elaboración de lo explicado en el alcance, ya que para probar lo implementado no hace falta su implantación en alguna empresa en particular, sino que será de uso web para un público en general, por lo que bastará con mostrar su uso vía un explorador. Por otro lado, existe un riesgo en cuanto al tiempo en que se emplee en la implementación del sistema de información web, debido a que podría sobrepasarse el tiempo planificado.

8 Viabilidad y justificativa del proyecto

La viabilidad de la elaboración del proyecto dependerá de los siguientes aspectos:

- Por un lado, se encuentra el aspecto técnico, para este se realizará uso de bases de datos, lenguajes, frameworks y servicios web gratuitos y bien documentados para que el acceso a ellos sea fácil y exista apoyo en cuanto a su uso a lo largo de la implementación. Asimismo, se hará uso de los papers listados en la bibliografía para el desarrollo de los algoritmos elegidos.
- Por otro lado, en cuanto a las metodologías, se utilizará la bibliografía citada anteriormente, la cual se brinda en la universidad, lo cual facilitará su consulta durante el desarrollo del proyecto de tesis.
- Por otro lado, en cuanto al tiempo que se dedicará a la elaboración del proyecto, se tendrá aproximadamente seis meses para su desarrollo desde el análisis hasta la implementación, lo cual es un tiempo prudente para abarcar dichas etapas en su totalidad para el sistema que se plantea elaborar.
- Finalmente, la necesidad de la elaboración de este proyecto se debe a que resuelve la problemática mencionada inicialmente: la falta de centralización de restaurantes y la falta de medios de comunicación entre restaurantes y clientes.

En cuanto a la justificación del proyecto, se menciona lo siguiente:

Por un lado, en cuanto a la motivación del desarrollo de este proyecto se encuentra lo siguiente:

- Frente al estudio del estado del arte, se identificó algunos puntos que podrían ser presentados como ventajas frente a los productos ya existentes. Por un lado, el sistema integrará a los consumidores y a los restaurantes en un mismo espacio para que se agilice la comunicación entre ellos. Por otro lado, el producto que se elaborará hará uso de un algoritmo que permitirá analizar los comentarios realizados sobre los restaurantes para controlar su contenido.

- En cuanto al tipo de proyecto (análisis, diseño e implementación), se eligió por el interés en el aprendizaje del desarrollo completo de un software web y por la experiencia en el uso de herramientas de desarrollo.

Por otro lado, en cuanto a los beneficios que la finalización del proyecto en su totalidad brindará consigo se encuentran las siguientes:

- Permitirá brindar un prototipo que muestre cómo los consumidores pueden ubicar con mayor facilidad restaurantes que se adecúen a sus necesidades a través de un espacio que los centralice. Este prototipo brindaría un punto de partida para futuros proyectos comerciales.

9 Plan de actividades

A continuación se muestra el cronograma final que se realizó para el cumplimiento del proyecto, este fue sometido a cambios debido a las revisiones y correcciones sobre los artefactos elaborados en las distintas fases que se muestran en el plan de actividades. Asimismo, para las fase de Concepción se ha estimado un promedio de tres horas de trabajo por cada día indicado; para la fase de Elaboración, se ha estimado un promedio de cuatro horas; y para la fase de Construcción, un promedio de 8 horas. Esto daría un total de 186 horas, 184 horas y 256 horas respectivamente. En un proyecto normal que siga RUP, la fase de Concepción tiende a ser mucho menor en horas invertidas que las fases de Elaboración y Construcción; sin embargo, para este proyecto se formó el cronograma de esta manera debido a las constantes correcciones que se realizaron durante la definición de la problemática a resolver. En cuanto a la fase de Construcción, a esta se le tuvo que invertir mayor cantidad de horas diarias debido a que el plazo límite de entrega del prototipo funcional se encontraba cerca.

En cuanto a los hitos de las iteraciones, en la fase de Concepción, se organizó solo una iteración que tendría como hito final la **aceptación del proyecto** que se obtendría por parte de los jurados de la sustentación final del curso de tesis 1. En cuanto a la fase de Elaboración, el hito de la segunda iteración corresponde a la obtención de una **arquitectura preliminar ejecutable**. Esta permitiría que se mitiguen riesgos técnicos

para evitar la aparición de retrasos durante la implementación del sistema. Por otro lado, el hito de la tercera iteración corresponde a obtener un **detalle sólido del sistema** a implementar, es por esto que la mayoría de sus actividades corresponden al desarrollo de los artefactos de análisis y diseño. En cuanto a la fase de Construcción, el hito de la cuarta iteración corresponde a la obtención de una **versión parcial del prototipo funcional del sistema de información**. Finalmente, la quinta iteración corresponde a la obtención de una **versión completa del prototipo funcional**.

Como se puede observar en el cronograma, se ha seguido el carácter iterativo e incremental de RUP, ya que los documentos o artefactos se han ido modificando a lo largo de las distintas fases. Entre algunos de ellos está la visión del proyecto que se fue mejorando a través de las distintas sustentaciones piloto o preliminares, los diagramas de clases y de base de datos durante la fase Construcción, entre otros.



3. Construcción	32 días	mar 07/05/13	mié 19/06/13
Iteración 4	7 días	mar 07/05/13	mié 15/05/13
Desarrollo de interfaces	3 días	mar 07/05/13	jue 09/05/13
Desarrollo de plan de pruebas	2 días	vie 10/05/13	lun 13/05/13
Paquete de Seguridad	1 día	mar 07/05/13	mar 07/05/13
Desarrollo del inicio de sesión	1 día	mar 07/05/13	mar 07/05/13
Paquete de Restaurantes	2 días	mar 07/05/13	mié 08/05/13
Desarrollo de registro de restaurante	1 día	mar 07/05/13	mar 07/05/13
Desarrollo de administración de información de restaurante	1 día	mié 08/05/13	mié 08/05/13
Paquete de Administración	2 días	mar 07/05/13	mié 08/05/13
Desarrollo de administración de servicios	1 día	mar 07/05/13	mar 07/05/13
Desarrollo de administración de tipos de locales	1 día	mar 07/05/13	mar 07/05/13
Desarrollo de administración de usuarios restaurantes	1 día	mié 08/05/13	mié 08/05/13
Paquete de Comensales	3 días	mar 07/05/13	jue 09/05/13
Desarrollo de la consulta de restaurantes	1 día	mar 07/05/13	mar 07/05/13
Desarrollo de la consulta de detalle de restaurante	1 día	mié 08/05/13	mié 08/05/13
Desarrollo de criticar restaurante	1 día	jue 09/05/13	jue 09/05/13
Ejecución de pruebas	1 día	vie 10/05/13	vie 10/05/13
Desarrollo completo del modelo 4+1 Vistas de Arquitectura	1 día	lun 13/05/13	lun 13/05/13
Desarrollo de estándares de programación	1 día	mar 14/05/13	mar 14/05/13
Revisión de versión parcial del sistema con el asesor	1 día	mié 15/05/13	mié 15/05/13
Iteración 5	25 días	jue 16/05/13	mié 19/06/13
Paquete de Seguridad	4 días	jue 16/05/13	mar 21/05/13
Integración de Authentication Forms	2 días	jue 16/05/13	vie 17/05/13
Corrección del inicio de sesión	1 día	lun 20/05/13	lun 20/05/13
Desarrollo de la identificación de roles	1 día	lun 20/05/13	lun 20/05/13
Desarrollo del cambio de contraseña	1 día	lun 20/05/13	lun 20/05/13
Desarrollo de la configuración límite de intentos fallidos	1 día	mar 21/05/13	mar 21/05/13
Desarrollo del desbloqueo de acceso	1 día	mar 21/05/13	mar 21/05/13
Paquete de Restaurantes	3 días	mié 22/05/13	vie 24/05/13
Corrección de registro de restaurante	1 día	mié 22/05/13	mié 22/05/13
Corrección de administración de información de restaurante	1 día	mié 22/05/13	mié 22/05/13
Desarrollo de administración de catálogo c	1 día	jue 23/05/13	jue 23/05/13
Desarrollo de consultar críticas	1 día	jue 23/05/13	jue 23/05/13
Desarrollo de consultar quejas	1 día	vie 24/05/13	vie 24/05/13

Desarrollo de reportar quejas	1 día	vie 24/05/13	vie 24/05/13
Paquete de Administración	6 días	lun 27/05/13	lun 03/06/13
Corrección de administración de servicios	1 día	lun 27/05/13	lun 27/05/13
Corrección de administración de tipos de locales	1 día	lun 27/05/13	lun 27/05/13
Corrección de administración de usuarios restaurantes	1 día	lun 27/05/13	lun 27/05/13
Desarrollo de administración de tipos de cocina	1 día	mar 28/05/13	mar 28/05/13
Desarrollo de administración de medios de pago	1 día	mar 28/05/13	mar 28/05/13
Desarrollo de administración de críticas de restaurantes	1 día	mar 28/05/13	mar 28/05/13
Desarrollo de administración de quejas de restaurantes	1 día	mié 29/05/13	mié 29/05/13
Desarrollo de administración de usuarios comensales	1 día	mié 29/05/13	mié 29/05/13
Modificación de la base de datos	1 día	mié 29/05/13	mié 29/05/13
Corrección del diagrama de base de datos	1 día	mié 29/05/13	mié 29/05/13
Adaptación del algoritmo Fuerza Bruta	1 día	jue 30/05/13	jue 30/05/13
Desarrollo de la gestión de críticas automatizada	2 días	vie 31/05/13	lun 03/06/13
Paquete de Comensales	5 días	mar 04/06/13	lun 10/06/13
Integración con el mecanismo de geolocalización de Google	2 días	mar 04/06/13	mié 05/06/13
Adaptación del algoritmo Shell	1 día	jue 06/06/13	jue 06/06/13
Desarrollo de la consulta de restaurantes	1 día	vie 07/06/13	vie 07/06/13
Desarrollo de la consulta de detalle de restaurante	1 día	vie 07/06/13	vie 07/06/13
Desarrollo de criticar restaurante	1 día	lun 10/06/13	lun 10/06/13
Ejecución de pruebas	1 día	mar 11/06/13	mar 11/06/13
Sustentación preliminar	1 día	mié 12/06/13	mié 12/06/13
Corrección del capítulo 1	4 días	jue 13/06/13	mar 18/06/13
Entrega de prototipo funcional del sistema con el asesor	1 día	mié 19/06/13	mié 19/06/13

10 Marco teórico

El objetivo de la revisión del marco teórico es poder detallar los conceptos que se encuentran relacionados con la problemática y, así, permitir entender mejor lo que se está queriendo solucionar en este proyecto de fin de carrera. A continuación se presenta el marco conceptual del problema.

10.1 Crecimiento de la gastronomía en el Perú

Durante el 2009, 10 mil 118 licencias para la apertura de restaurantes se otorgaron a nivel de todo el país, lo cual indica que se dieron 28 licencias por día (INEI 2011). Asimismo, haciendo una comparación del sector restaurantes entre el 2009 y el 2010, esta ha experimentado un crecimiento del 4.76% (INEI 2010). Por otro lado, según el presidente del Subcomité de Gastronomía de la Cámara de Comercio de Lima, el 50% de los restaurantes, que mensualmente se inauguran, fracasan.

Estas cifras muestran un movimiento bastante dinámico en el sector mencionado, lo cual hace muy difícil poder tener acceso a información sobre los distintos restaurantes a nivel de todo el Perú.

10.2 Competencia entre restaurantes

La visión tradicional acerca de la competencia entre restaurantes significaba que si un nuevo local aparecía cerca de otro, este era el inicio de un conflicto o riesgo para el que ya estaba establecido. Esta visión de la competencia ha cambiado a lo largo del tiempo. Hoy en día, a la competencia se le ve como una oportunidad de mejora. “... use the competition to improve yourself” (Willis 2012) es lo que menciona Willis y afirma es lo que está ocurriendo en la actualidad. En el caso de nuestro país, se comprueba con la cantidad de restaurantes que abren y cierran en el mercado anualmente.

Tomando como referencia este nuevo enfoque de los restaurantes, la guía gastronómica poseerá las características de poder calificar y comentar, lo cual le permitirá a los restaurantes sobresalir por la calidad de sus servicios, y así se demostrará

lo que mencionaba Willis de usar la competencia (y en este caso las críticas) para mejorar los servicios que brinda el restaurante.

10.3 Calidad del servicio en restaurantes

Los distintos negocios existentes necesitan poder retener clientes rentables para la continuidad de este. Esto se obtiene a través de dos conceptos claves: la marca y el posicionamiento. Es importante que se tenga un posicionamiento diferenciado para que el cliente no vea a la marca como a las demás, sino que la elija sobre las otras (IPM 2011). En el caso de los restaurantes, este posicionamiento se obtiene a través de la calidad del servicio (la buena atención y la calidad de la cocina principalmente) que van construyendo a lo largo de su existencia en el mercado.

En la actualidad existen restaurantes que no invierten en su marca ni su posicionamiento, probablemente por el tiempo y la inversión que esta actividad puede significar. De esta forma, por un lado, es difícil para los clientes poder obtener un conocimiento a priori de la calidad de los servicios en los restaurantes; y por otro, los restaurantes disminuyen su rentabilidad.

En la guía gastronómica se pretende crear espacios para los restaurantes donde el público pueda comentar, y así ir mostrando la calidad en los servicios que los clientes reciben. De tal forma que se va creando una imagen de los restaurantes, es decir, se irá trabajando en el posicionamiento de cada restaurante de una forma ágil.

10.4 Portales Web como canales de publicidad para restaurantes

Los portales web han sido una herramienta ampliamente usada por distintos negocios para poder llegar a sus clientes debido a la cantidad relevante de personas que se encuentran navegando por internet.

En Perú, cada año se está experimentando un incremento en cuanto a la población que posee acceso a internet, ya sea desde su hogar, cabinas de internet, móviles, etc. A

finales del 2012, se registró que el 22.4% de los hogares poseían acceso a internet con un aumento de 4.6 puntos con respecto al año anterior. Asimismo, se indicó que el 39.4% de la población de 6 a más años de edad hace uso de internet con un aumento de 1.9 puntos con respecto al año anterior (INEI 2013). Así, si se toma como referencia el censo nacional del 2008, aproximadamente 6,140,323 de personas es la cantidad a la que se puede tener acceso por medio de este canal (INEI 2008).

Así, los restaurantes han sido otro negocio más que ha hecho uso de portales web para poder llegar a más clientes. Sin embargo, como se mencionaba en la problemática, no todos los restaurantes poseen un portal web para poder ser ubicados, y por otro lado, la información que muestran (en la mayoría de los casos) es publicada por ellos mismos, lo cual no permite tener un conocimiento real de los servicios que brinda por parte de los clientes.

10.5 Posicionamiento de Portales Web

SEO se refiere a Posicionamiento y Optimización en Sitios Web, la cual es una actividad para que un sitio web sea rentable a través del incremento en visitas hacia tu web (AGUILAR 2011: 15). El SEO posee unos principios básicos que de ser seguidos por un sitio web, este será más sencillo de ubicar por el público. Esta actividad requiere de tiempo e inversión en una organización, ya que involucra poseer conocimiento sobre los clientes, la industria, programación y seguir algunas reglas propias del SEO.

El SEO es una actividad que también debe abarcarse en el desarrollo de portales web de restaurantes, y como se mencionó, es una inversión, la cual puede no ser efectuada por algunos restaurantes, ya sea por falta de recursos, o porque no logran entender los beneficios que esta brinda. Esto genera problemas para los restaurantes, ya que no les permite ser ubicados fácilmente por consumidores que podrían estar interesados en sus servicios. Por lo tanto, con la guía gastronómica se busca hacer transparente esta actividad para los restaurantes, ya que todos los restaurantes estarán organizados en un mismo sitio web.

10.6 Social Media Marketing en restaurantes

Los medios sociales en el marketing permiten a las empresas y los consumidores interactuar a través de comentarios. Estos implican un cambio del marketing tradicional, en el cual los consumidores son solo lectores, a un entorno en el cual tanto los autores como el público interactúan compartiendo información (Evans 2008: 32). Ejemplos de medios sociales son Facebook, Foursquare, Youtube, Wikipedia, en los cuales se puede apreciar esta característica de la interacción entre autores y la gente compartiendo información a través de comentarios. Este sistema de marketing brinda una “voz” a los consumidores, lo cual permite a las empresas obtener un feedback, lo cual los ayuda a detectar cómo mejorar los servicios que brindan; sin embargo, esta característica trae consigo un control limitado sobre lo que el público comenta, lo cual crea un temor en las organizaciones por los mensajes negativos que se puedan generar.

Este marketing por medios sociales se comenzó a efectuar en los restaurantes cuando estos invirtieron en portales web con espacios donde los consumidores pudiesen comentar sobre sus servicios. Lo que se busca en la solución es que estas características de medios sociales se mantengan, pero ahora en una guía gastronómica que poseerá características de una red social, en la cual los restaurantes tendrán secciones para comentarios y calificaciones por parte de los consumidores. Así, será esencial el rol de estos últimos, ya que será en base a ellos que se genere información pertinente en cuanto a la calidad de los servicios que brindan los restaurantes. Asimismo, esta solución busca controlar lo que el público comenta para que no se caiga en mensajes negativos que solo busquen desprestigiar a algún restaurante dentro del contexto de la libre competencia.

10.7 Factores importantes en los costos de un restaurante

Existen ciertos factores que de ser mal manejados, pueden influenciar negativamente en los costos de un restaurante. Factores tales como la selección del proveedor, pedido de la mercancía, recepción de la mercancía, no se acomoda adecuadamente la mercancía, temperaturas de refrigeración o congelación, falta de control en el almacén, falta de control en las requisiciones, falta de control en áreas de

producción, exceso o falta de producción, errores en las tomas de orden a los clientes y cobro al cliente (Cuevas 2004: 53). Se hace énfasis en el error en la toma de órdenes, ya que esto no solo afectará a la elaboración de platillos innecesarios, sino que también incomodará al cliente, ocasionándole una opinión negativa hacia el negocio. Esto último, influye en el posicionamiento de un restaurante, ya que afecta su imagen directamente.

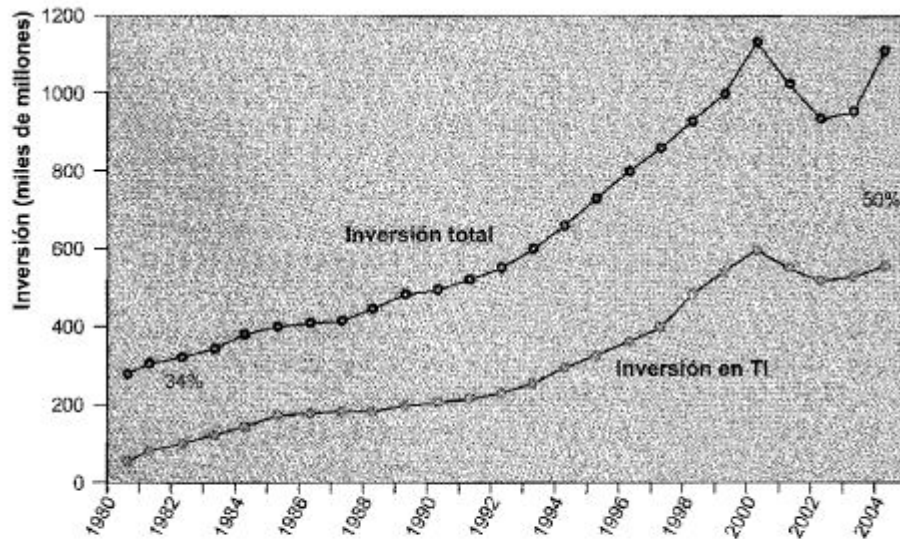
Este concepto se relaciona con el problema por el hecho de lo importante que son estos comentarios que reflejan el servicio brindado por un restaurante, y lo importante que es que tanto el restaurante como el cliente lo tengan claro. Desde la perspectiva del consumidor es importante para conocer al restaurante; y desde la perspectiva del restaurante es importante para obtener un feedback de sus servicios.

10.8 Sistemas de Información: Guías Gastronómicas

Ya desde estos últimos años, se puede decir que no hay negocio que no interactúe con software. "En el 2006, las empresas estadounidenses gastaron 1.8 billones de dólares en hardware, software y equipo de telecomunicaciones para los sistemas de información" (Laudon y Laudon 2008: 5). Para tener un poco más de conocimiento acerca de cómo las empresas han ido adaptando los sistemas de información como parte importante de la función de su negocio, en la figura 1.2, se muestra como creció la inversión de capital en tecnología de información durante el periodo de 1980 al 2004. Todo lo anterior, coloca a las empresas en un estado de alerta de cómo juntar la tecnología con el negocio, ya que existen muchos procesos que se pueden optimizar a través de la automatización. En base a lo anterior, es importante la decisión de la tecnología de la que se hará uso para un negocio en particular, porque como ya se demostró, gran parte del capital va dirigido a esta en distintos negocios a nivel mundial.

En el caso de los restaurantes, existen distintos sistemas de información que se han aplicado para la administración de estos. Uno de estos son las guías gastronómicas, las cuales (a diferencia de sistemas particulares para cada restaurante) no son implantadas en un local en particular, sino que son espacios públicos en los cuales los restaurantes pueden publicar información acerca de ellos y los clientes pueden buscarlos, criticarlos

y calificarlos para así poder armar un espacio con amplio contenido acerca de los distintos locales en una región.



La inversión de capital en tecnología de información, definida como hardware, software y equipo de comunicaciones, creció de 34 a 50% durante el periodo de 1980-2004.

Fuente: Con base en datos del Departamento de Comercio, la Oficina de Análisis Económicos y las Cuentas Nacionales de Ingresos y Productos de Estados Unidos, 2006.

Figura 1.2 "Crecimiento de inversión de capital en TI"

10.9 Consideraciones para implementar TI en un restaurante

Enfocarse en una tecnología para ayudar al negocio, no es un resultado, sino un proceso, ya que los negocios deberán estar constantemente verificando las tecnologías nuevas que puedan brindarle ayuda. Además, se debe diferenciar entre los impactos funcionales y operativos que trae consigo las TI, en este caso, las TI pueden afectar toda la organización desde sus procesos. Además, se debe realizar un análisis financiero, enfocarse en el manejo de la información y considerar los tiempos de la implantación (ORONSKY, Carl y Prakash CHATHOTH: 2007). Esto debido a que teniendo en cuenta estas tareas, se tendrá una mayor seguridad de que optar por el uso de TI, será la mejor opción.

En el caso de la guía gastronómica, esta no es una herramienta que se vaya a implementar en los restaurantes como un sistema de información personalizado, sino que será una herramienta de TI externa de la cual los restaurantes podrán aprovechar su uso para obtener feedback. Las consideraciones para este tipo de herramientas de TI

será el tiempo que se le dedique y cómo afectará a mejorar los procesos de la empresa, en este caso, la calidad de los servicios.

10.10 Mecanismos de geolocalización en entornos web

Los mecanismos de geolocalización se han vuelto bastante usados en dispositivos móviles como los smartphones. Estos tienen la ventaja de contar con sistemas GPS integrados. Sin embargo, empresas como Google utilizan estos mecanismos para dispositivos que no cuentan con sistemas GPS integrados como son los navegadores de internet. Los exploradores de internet como Chrome, Firefox, y otros en sus últimas versiones, soportan la geolocalización a través de las librerías de Google Maps que la empresa en cuestión brinda. Gracias a este aporte, se ha hecho posible contar con la posición de los usuarios conectados a una red de internet.

Uno de los aportes que se quiere realizar con este proyecto de fin de carrera es hacer uso de este servicio de Google para poder comparar las posiciones con los restaurantes registrados en el sistema, y así mostrar las distancias de los usuarios con respecto a los restaurantes, de tal forma que estos puedan evaluar si los restaurantes se ajustan a sus necesidades en cuanto a distancia.

11 Estado del arte

11.1 Objetivo de la revisión

El objetivo de la revisión del estado del arte es el análisis de algunas funcionalidades de sistemas de información existentes que han buscado resolver el problema tanto de forma exacta como aproximada. Así, se podrá tomar como base dichos aspectos para no tener que partir desde cero en el análisis del sistema que se propone como solución en este proyecto.

- *Degusta*

Degusta es una plataforma que funciona a modo de red social con el objetivo de brindar, a sus usuarios, información acerca de restaurantes en el Perú (esto es en el caso de “Degusta Perú”, ya que tiene distintas versiones para cada país). La información que brinda con respecto a los restaurantes, y que es relevante para la solución de la problemática, es la **clasificación global**, la **cantidad de votos**, los **servicios** que brindan, los **platos** que sirven, el tipo de cocina y su ubicación (DEGUSTA 2012).

Esta plataforma cuenta con una aplicación web tanto como móvil. En las figuras 1.3 y 1.5, se puede visualizar una pantalla de la versión web en la que se consulta información de un restaurante (La Lucha en este caso). En esta se muestran datos relevantes como la calificación global, la cantidad de votos por parte de los usuarios y los servicios que el local brinda (DEGUSTA 2012).



The screenshot shows the Degusta website interface for the restaurant 'La Lucha'. The header includes the Degusta logo and navigation links: Inicio, Restaurantes, and Mi degusta. The user profile 'CHRISTIAN P. | SALIR' is visible in the top right corner. The main content area displays the restaurant's details:

- restaurante** (category)
- La Lucha** (name)
- Av. Diagonal 308 - Miraflores - Lima
- Tel.: 241 5953
- Tipo: Cafetería
- Cocina: Sandwiches
- Medios de pago: Efectivo | 
- Horarios:
 - Lunes a Jueves de 6:00 a.m. a 1:00 a.m.
 - Viernes y Sábados de 6:00 a.m. a 3:00 a.m.
 - Domingos de 6:00 a.m. a 1:00 a.m.
- Ver Sucursales
- Calificaciones de los usuarios:

Comida	Servicio	Ambiente	Precio	Votos
4.1/5	3.7/5	3.4/5	S/. 21	22

Below the ratings table, there is a link to 'Sobre las calificaciones y los precios'. To the right, there is a map section titled 'UBICACIÓN' showing the restaurant's location in Miraflores, Lima, and a 'SERVICIOS' section listing 'Acceso para discapacitados'.

Figura 1.3 "Degusta Detalle Restaurante I" (DEGUSTA 2012)

Asimismo, en la figura 1.5 se muestra la sección final de las dos figuras mostradas anteriormente, la cual contiene información acerca de los platos más recomendados del local y comentarios de los usuarios (DEGUSTA 2012).

Comida	Servicio	Ambiente	Precio	Votos
4.1/5	3.7/5	3.4/5	S/. 21	22

Sobre las calificaciones y los precios



Fotos de los usuarios:



SERVICIOS

- Acceso para discapacitados
- Baños para discapacitados
- Preferible reservar
- Música en vivo
- Menu ejecutivo
- Estacionamientos
- Valet Parking
- Buffet
- Wi-fi
- Barra
- Envío a domicilio
- Descorche
- Menu para niños
- Carta de vinos
- Mesas al aire libre
- Desayunos

Figura 1.4 “Degusta detalle restaurante II” (DEGUSTA 2012)

Platos recomendados:

- Sanguche la lucha 10
- Papas huayro fritas 8
- Sanguche de pavo a la leña 7

[Ver más](#)

Comentarios

+ Calificar Restaurante

22 comentarios

ORDENADOS POR: [mas reciente](#) | mejor calificación

DEL 1 AL 8

VER VOTOS: [filtrados](#) | malos | regulares | buenos | muy buenos | excelentes | todos

Marco D.

64 puntos
5 comentarios
0 amigos
Conocedor

Publicado desde iPhone App: 18/09/2012

Para mi la mejor sangucheria de lima. Sus panes son increíbles, el de lechón y el de pavo son mis favoritos. El arma secreta de este local son sus papas huayro con piel que son exquisitas (pruebenlas con las salsas para vivir una experiencia inimaginable).

Comida ★★★★★ Servicio ★★★★★ Ambiente ★★★★★

Ana Paula P.

95 puntos
9 comentarios
0 amigos
Gastrónomo

Publicado: 17/09/2012

Figura 1.5 “Degusta detalle restaurante III” (DEGUSTA 2012)

Por otro lado, se encuentra la versión móvil. En las figuras 1.6 y 1.7, se puede apreciar un par de ventanas de la aplicación en iOS y cómo muestran la información mencionada en la versión web (calificación, votos, platos recomendados, comentarios, entre otros). En estas, la diferencia, con respecto a la versión web, es la forma en que se muestra la información por temas de experiencia de usuario, pero dicha información sigue siendo la misma.

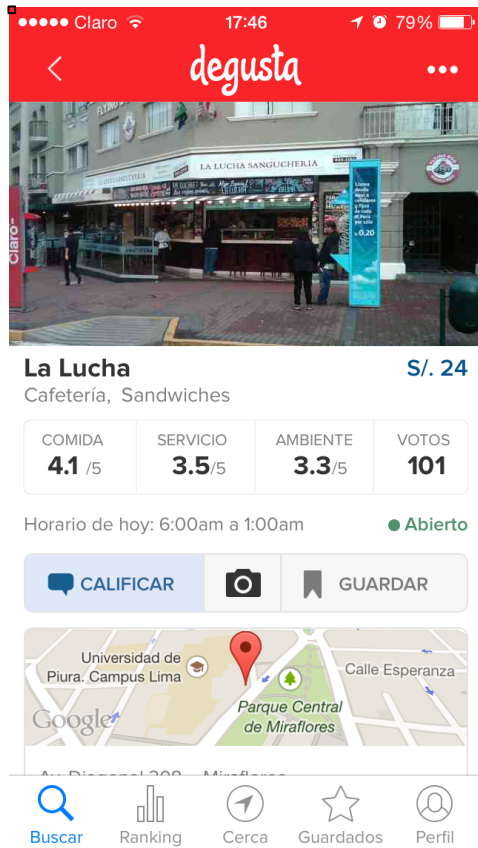


Figura 1.6 "Degusta Móvil – Restaurante I" (DEGUSTA 2015)



Figura 1.7 "Degusta Móvil - Restaurante II" (DEGUSTA 2015)

Asimismo, una diferencia relevante en la versión móvil se encuentra en su opción llamada "cerca" (la cual no se encuentra en la otra versión). Esta muestra un listado en orden de cercanía de los restaurantes con su respectiva distancia gracias al GPS que incorporan los celulares inteligentes. A su vez, el listado se puede mostrar en un mapa como se puede apreciar en las figuras 1.8 y 1.9.

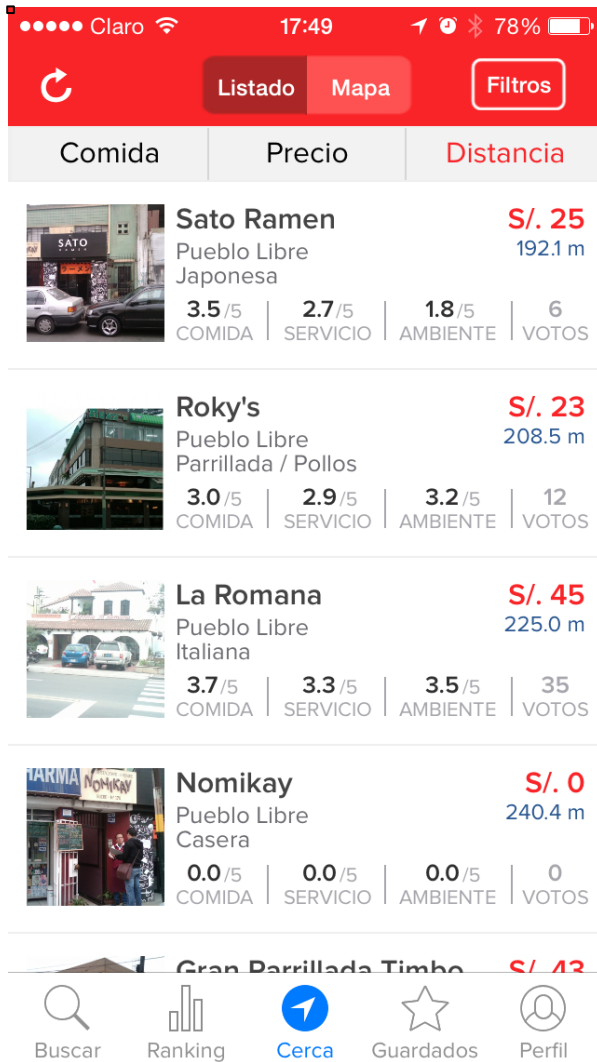


Figura 1.8 "Degusta Móvil - GPS I"
(DEGUSTA 2015)

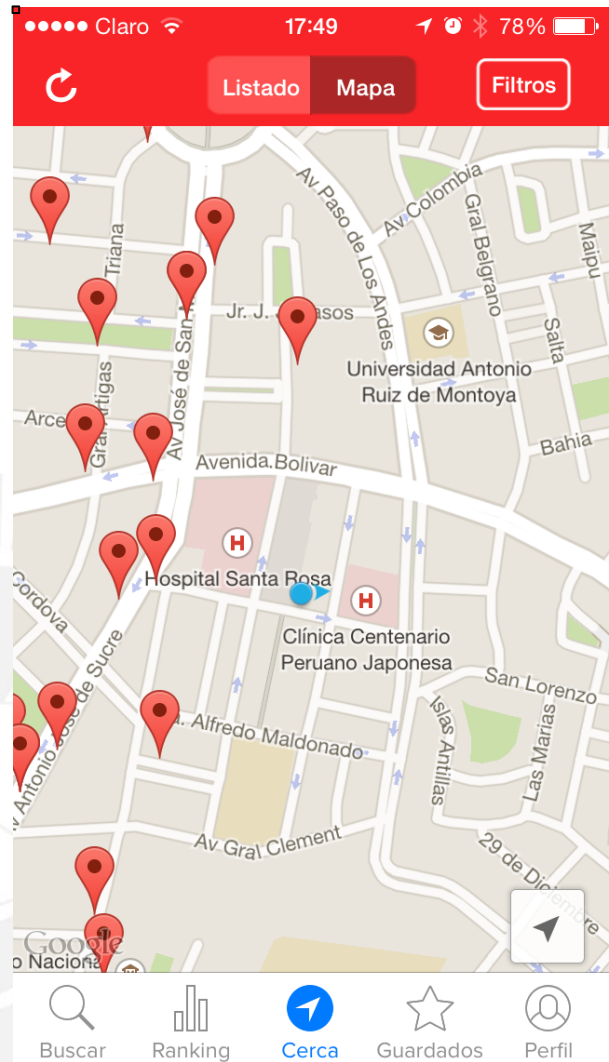


Figura 1.9 "Degusta Móvil - GPS II"
(DEGUSTA 2015)

- **GPS restaurante**

GPS restaurantes es una guía de restaurantes de Lima y a nivel nacional (GPS RESTAURANTES 2012). Esta guía cuenta con la característica de poder comentar, es decir, realiza un feedback hacia los servicios de los restaurantes. Asimismo, posee varios criterios de búsquedas como por tipo de comida, ubicación, precios, entre otros.

En la figura 1.10, se puede visualizar su página principal, en la cual se puede ver su principal buscador que divide la información en base a tipos de comida, ubicación y precios promedio. Estos filtros mencionados para la búsqueda son un tanto limitados, a

comparación de Degusta; sin embargo, es un guía constantemente actualizada, por lo tanto relevante para Lima.



Figura 1.10 “GPS Restaurantes.pe” (GPS RESTAURANTES 2012)

- **Guía Oleo**

Guía Oleo es una guía gastronómica que busca centralizar y organizar la información de restaurantes en Argentina para que consumidores interesados ubiquen el restaurante que se adecúe a sus necesidades. Asimismo, brinda espacios en los cuales se puede comentar acerca de un restaurante en específico, de tal forma que Guía Oleo también cuenta con la característica de realizar feedback sobre sus servicios brindados, y desde la perspectiva de los consumidores, de saber qué tan bueno realmente es el servicio que cada restaurante brinda. (GUÍA OLEO S.A.: 2012).

En la figura 1.11 y 1.12, se puede visualizar la interfaz en la cual Guía Oleo brinda un espacio a cada restaurante para publicar información acerca de ellos y comentar sobre sus servicios. Además, se puede visualizar detalles como cuántos usuarios recomiendan el restaurante, los servicios que brinda el restaurante, e incluso posee una opción para realizar reservas.

Francesco

Soler 5598, [Palermo](#) > [Norte JB Justo](#), C.A.B.A.

Cocina: [Peruana](#)

[Reservá tu mesa](#)

Tel: [4774-4011](#) / 1537959634

Horario: Hoy 12 a 15 y de 20 a 24

Medios de pago: Otros medios de pago

[Reportar cierre/modificación](#)

Guardar... +1 0

Comida: 26 /30 Servicio: 20 /30 Ambiente: 23 /30 Carta de vinos: Buena

Calificá este restaurante:

Lo recomiendo No lo recomiendo

25 votos (18 comentarios) • [Ver histórico de calificaciones](#)

Chef: Percy Quincho Segura

Especialidades: Ceviche Mero a lo macho Suspiro a la limeña



Restaurantes similares destacados

[El Estanciero](#)

\$186 alm / \$186 cena

Recomendado para:

Ir en pareja 35 usuarios

Reunión de negocios 12 usuarios

Comer con buenos tragos 9 usuarios

[ver más](#)

Características y restaurantes similares

- Acceso discapitados
- Baños discapitados
- Reservas
- Show
- Mesas al aire libre
- Estacionamiento
- Juegos para chicos
- Delivery
- Área fumador
- Aire acondicionado
- Barra de tragos
- Menú ejecutivo
- Descorche



Difundir Francesco

URL corto: <http://vinos.guiaoleo.com/u/11570>

Difundir:



Figura 1.11 “Guía Oleo I”(Guía Oleo 2012 S.A.)

18 Comentarios Ordenados por fecha o por [experiencia de usuario](#)

Ver votos: Todos, [excelentes](#), [buenos/muy buenos](#), [regulares/malos](#)

Enrique5411 ★★★★★
 12 Oct 2012 Eramos dos parejas y fuimos a cenar un viernes. Comida: buena calidad y presentación, la cantidad escasa. Servicio: Malo, el mozo estaba disperso y no sabia explicar adecuadamente los platos. Olvido o no entendió el pedido de la entrada y ordeno solo los platos principales. Ambiente muy ruidoso. El vino que ordenamos no estaba en stock tuvimos que cambiar por otro. Habiendo llegado a las 21:15, esperamos mas de una hora por el plato principal y siendo las 12:30 comenzaron a demostrar interés en que nos vayamos. No volvería.
 Comida: Servicio: Ambiente:
 Muy útil 1 voto • [Responder](#) • [Reportar indebido](#)

aquiles2580 ★★★★★
 28 Sep 2012 demasiado caro para un servicio tan escaso, muy mal atendidos y solo les interesaba el momento del cierre
 Comida: Servicio: Ambiente:
 Muy útil • [Responder](#) • [Reportar indebido](#)

TuchoP ★★★★★
 28 Sep 2012 De lo mejor para comer comida peruana. Los platos son todos muy ricos, y la calidad de los alimentos es altísima. El servicio es realmente bueno, la atención de los mozos es perfecta, cero invasivos pero siempre atentos por si necesitas algo. Lo recomiendo enormemente. (en especial los tiraditos tricolor, el lomo a la huancaina y el suspiro limeño)
 Comida: Servicio: Ambiente:
 Muy útil 1 voto • [Responder](#) • [Reportar indebido](#)

Figura 1.12 “Guía Oleo II”(Guía Oleo 2012 S.A.)

- **TripAdvisor**

TripAdvisor Inc. parte de la problemática de que existen muchos lugares, restaurantes y hoteles sobre los cuales se puede trazar un plan de viaje. En base a esta problemática, se desarrolla TripAdvisor, la cual es una aplicación en entorno web que recopila información de lugares turísticos, restaurantes y hoteles en todo el mundo para tener un acceso fácil y ágil a esta información y poder planificar un buen viaje (TRIPADVISOR INC. 2012). En la actualidad, su sistema cuenta con 74 millones de visitas mensuales aproximadamente, lo cual demuestra lo útil que ha sido esta para el público en general. En la figura 1.13, se puede visualizar cómo su interfaz orienta al usuario a crear su propio viaje seleccionando los hoteles, vuelos a reservar, restaurantes a visitar, entre otros.

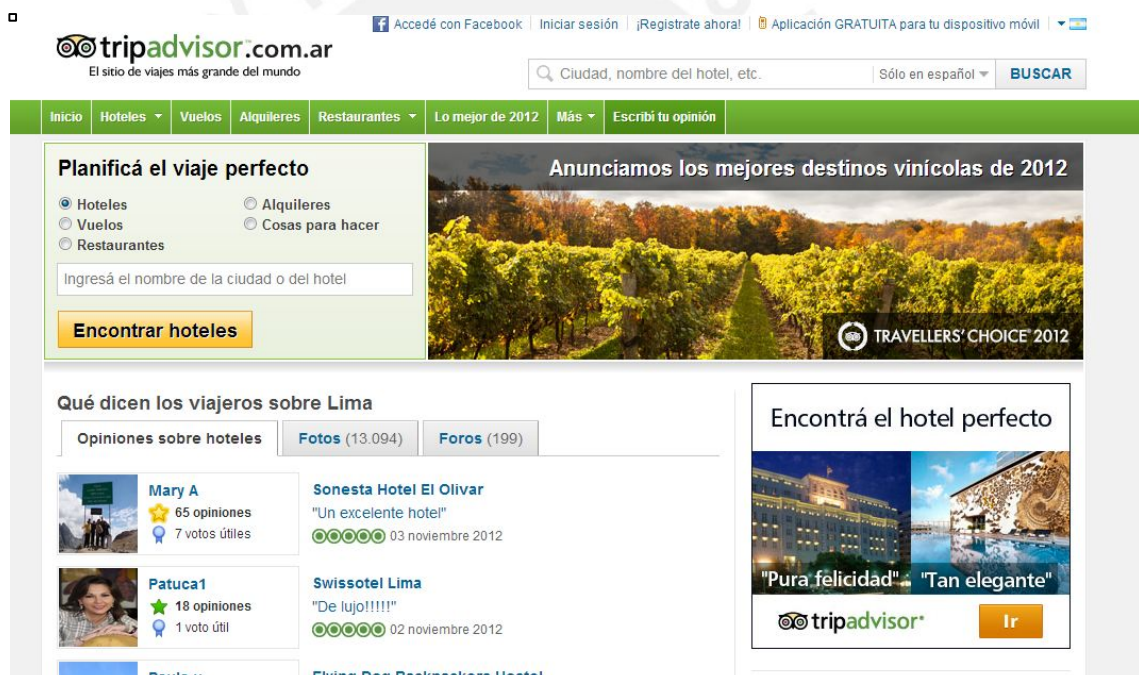


Figura 1.13 “TripAdvisor” (TRIPADVISOR INC. 2012)

- **QUEREMOSCOMER.COM**

QUEREMOSCOMER.COM es una de las guías gastronómicas más completas de la Ciudad de México. En esta se puede buscar restaurantes por criterios como la zona, precios, entre otros. Asimismo, ofrece la posibilidad de que cada nuevo restaurante cree su cuenta en el sistema y publique su información allí. Además, está en constante monitoreo de los usuarios del sistema para poder brindar opciones que podrían

agradarles en base a sus preferencias (QUEREMOSCOMER 2012). En la figura 1.14, se puede visualizar la interfaz para hacer uso de su búsqueda avanzada. Como se puede visualizar, posee campos como tipo de comida, zona y precios.



Figura 1.14 "QUEREMOSCOMER.COM" (QUEREMOSCOMER 2012)

- **Foursquare**

Foursquare es una aplicación a modo de red social que brinda ayuda a los usuarios para encontrar lugares cercanos recomendados por otros usuarios, es decir, los ayuda a encontrar algún lugar de su interés (gimnasios, restaurantes, colegios, etc.). Asimismo, Foursquare se centra en que dichos lugares (negocios en muchos casos) puedan tener un espacio para tener contacto con los usuarios que los visitan mostrando información (en el caso de tiendas podría querer mostrarse ofertas). Su principal mecanismo que permite que lo anterior funcione es el concepto de "check-in". Por medio de este, el usuario va registrando dónde ha estado y cuántas veces ha estado en un lugar en específico. Para mi problemática lo que es relevantes es el concepto de "check-in", ya que la lógica que usa sería útil para poder llevar un registro de lo que los clientes consumen (FOURSQUARE 2012).

Como se puede apreciar en las figuras 1.15, 1.16 y 1.17, primero se busca un lugar en particular, y luego puedes seleccionar la opción de "check-in". En la última imagen, se muestra el resultado de seleccionar dicha opción. En este caso, la aplicación muestra

que es la primera vez que el usuario hace check-in aquí, y se irá incrementando por cada check-in que se vuelva a registrar (FOURSQUARE 2012).

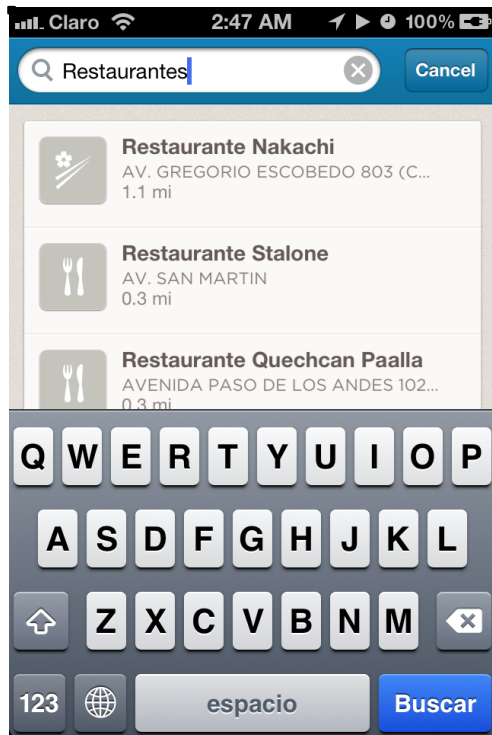


Figura 1.15 “Foursquare Búsqueda” (FOURSQUARE 2012)



Figura 1.16 “Foursquare Restaurante” (FOURSQUARE 2012)



Figura 1.17 “Foursquare check-in” (FOURSQUARE 2012)

11.2 Conclusiones sobre el estado del arte

A partir de las soluciones encontradas, se puede entender cómo el problema de no tener información de restaurantes, platillos y servicios centralizada y organizada se ha intentado resolver a través de guías gastronómicas en países como Argentina, México, Panamá y Perú. Las características que poseen las guías gastronómicas indicadas en esta sección son las de organizar los restaurantes por los servicios que brindan (buffet, desayuno, entre otros), precios, calificaciones, tipo de cocina, entre otros; brindar opciones de búsquedas en base a estos criterios; publicar novedades como nuevos restaurantes registrados, noticias sobre restaurantes ya existentes (como nuevos platillos, servicios, entre otros); y, por último, brindan un mecanismo por el cual los consumidores puedan interactuar con los restaurantes, es decir, publicar comentarios sobre ellos. En base a este resumen de características, lo que se busca en este proyecto, es brindar una guía gastronómica que tome como base estas características y ampliarlas; por un lado, utilizar las herramientas de geolocalización en entorno web existentes para poder hallar distancias entre los usuarios y los restaurantes para así poder ayudar a los usuarios a saber qué restaurantes son los más cercanos a ellos; por otro lado, realizar un método que permita dar control a los comentarios en el sistema para realizar un respectivo control de contenido y no se publiquen aquellos con mensajes malintencionados; por último, aprovechar el mecanismo de geolocalización para poder brindar búsquedas con filtros más detallados como sería el de distancia máxima con respecto al usuario.

CAPÍTULO 2: ANÁLISIS Y ARQUITECTURA

Este capítulo se centra en desarrollar y resolver el primero objetivo específico, el cual resulta en la especificación de la arquitectura web cliente servidor necesaria para el prototipo funcional. Para esto se empezará analizando el sistema de información para tener las necesidades especificadas y luego se elaborará la arquitectura necesaria para soportar dichas necesidades.

1 Identificación de requerimientos

Para el análisis, el primer paso a considerar será el de definir los requerimientos del sistema. Estos requerimientos serán agrupados en dos grupos: los funcionales y los no funcionales. Estos requerimientos se obtuvieron a través del previo estudio del marco conceptual y el análisis de las funcionalidades existentes en los sistemas mencionados en el estado del arte.

1.1 Requerimientos funcionales

Los requerimientos funcionales permitirán definir el comportamiento del sistema: input, output y el flujo de información. Con esta lista de requerimientos se podrá obtener las necesidades del presente software. En la siguiente lista se presentan los requerimientos agrupados en cinco grupos: "Seguridad", "Administración del Sistema", "Reportes", "Flujo Restaurantes" y "Flujo Consumidores". Esta misma organización permitirá más adelante organizar los casos de uso en paquetes. A continuación la lista:

Código	Descripción del Requerimiento
Administración del Sistema	
FUN-01	El sistema debe permitir que existan tres tipos de usuarios: restaurantes, clientes y administradores.
FUN-02	El sistema debe permitir que los usuarios restaurantes puedan publicar información sobre sus locales.
FUN-03	El sistema debe permitir a los usuarios clientes que critiquen a los restaurantes.
FUN-04	El sistema debe permitir a los administradores dar mantenimiento a

	los usuarios consumidores para poder darles de baja o reactivarlos.
FUN-05	El sistema debe permitir a los administradores dar mantenimiento a los usuarios restaurantes para poder darles de baja o reactivarlos.
FUN-06	El sistema debe permitir administrar los distintos servicios con los que un restaurante puede registrarse.
FUN-07	El sistema debe permitir administrar los distintos tipos de locales con los que un restaurante puede registrarse.
FUN-08	El sistema debe permitir administrar los distintos tipos de cocina con los que un restaurante puede registrarse.
FUN-09	El sistema debe permitir administrar los distintos medios de pago con los que un restaurante puede registrarse.
FUN-10	El sistema debe permitir que los administradores acepten o rechacen las críticas hacia los restaurantes antes de ser publicadas.
FUN-11	El sistema debe permitir que las críticas puedan ser aceptadas o rechazadas automáticamente a través un algoritmo que identifique texto con palabras malintencionadas.
FUN-12	El sistema debe permitir que los administradores acepten o rechacen las quejas reportadas por los restaurantes.
FUN-13	El sistema debe permitir que una vez aceptada una queja (crítica reportada), su crítica asociada ya no sea pública.
FUN-14	El sistema debe permitir consultar las solicitudes de nuevos restaurantes para poder registrarlos.
FUN-15	El sistema debe permitir generar reportes que muestren las críticas y quejas pendientes por responder.
FUN-16	El sistema debe permitir al administrador activar o desactivar el mecanismo automatizado que acepta o rechaza las críticas
FUN-17	El sistema debe permitir administrar las palabras que de ser encontradas en los comentarios, hará que estos no sean válidos (críticas rechazadas).
Flujo Restaurantes	
FUN-18	El sistema debe permitir enviar solicitudes para registrar nuevos restaurantes mediante una interfaz pública.
FUN-19	El sistema debe permitir a los restaurantes administrar su

	información a publicar.
FUN-20	El sistema debe permitir que los restaurantes tenga una página en la que puedan publicar toda la información que deseen sobre sus locales.
FUN-21	El sistema debe permitir que el restaurante registre su ubicación en un mapa.
FUN-22	El sistema debe permitir que el restaurante suba una foto de su local.
FUN-23	El sistema debe permitir que las páginas de los restaurantes tengan una sección de donde se publiquen las calificaciones y críticas de otros consumidores.
FUN-24	El sistema debe permitir a cada restaurante poseer su propio catálogo de platos y poder administrarlo.
FUN-25	El sistema debe permitir subir una imagen por cada plato que se registre junto con su información.
FUN-26	El sistema debe permitir a los restaurantes consultar las críticas que se hayan hecho sobre ellos.
FUN-27	El sistema debe permitir que los restaurantes reporten críticas (registren quejas) que consideren malintencionadas.
FUN-28	El sistema debe permitir a los restaurantes consultar sus quejas para saber si han sido aceptadas o no.
Flujo Consumidores	
FUN-29	El sistema debe permitir el registro de usuarios consumidores mediante una interfaz pública.
FUN-30	El sistema debe permitir a los usuarios consumidores criticar los restaurantes en base a la calidad del servicio, la comida, el ambiente y precio.
FUN-31	El sistema debe permitir que los usuarios registrados como consumidores puedan registrar comentarios criticando a los restaurantes.
FUN-32	El sistema debe permitir realizar búsquedas de restaurantes de forma pública sin necesidad de que se inicie sesión.
FUN-33	El sistema debe permitir realizar búsquedas con filtros que contengan “nombre del restaurante”, “servicios que brinda”, “rango de precios”,

	“tipo de restaurante”, “distrito en el que se ubica”, “tipo de comida” y “distancia con respecto al usuario”.
FUN-34	El sistema debe permitir mostrar los resultados de las búsquedas con la distancia entre el usuario y los locales, así como permitir ordenarlos en base a dicha distancia.
FUN-35	El sistema debe mostrar un mapa en el que se encuentre el restaurante consultado.
FUN-36	El sistema debe permitir a cualquier usuario (registrado o no) consultar restaurantes, su detalle y su ubicación.
FUN-37	El sistema debe permitir a los usuarios (registrados o no) consultar los detalles de los platos de los restaurantes.
FUN-38	El sistema debe permitir consultar todas las críticas hacia restaurantes que el comensal haya registrado.
Seguridad	
FUN-39	El sistema debe permitir a los distintos usuarios del sistema poder ingresar a este por medio de un inicio de sesión.
FUN-40	El sistema debe permitir bloquear el acceso a un usuario en caso este haya ingresado datos de autenticación erróneos una cantidad configurable de veces.
FUN-41	El sistema debe permitir que se configure la cantidad límite de intentos fallidos de autenticación.
FUN-42	El sistema debe permitir desbloquear el acceso a un usuario a través de una solicitud. Este debe responder de forma automática enviando una nueva contraseña al correo del usuario.
FUN-43	El sistema debe permitir a cualquier usuario poder cambiar su contraseña.

1.2 Requerimientos no funcionales

Los requerimientos no funcionales no proporcionarán mayor información sobre el comportamiento del sistema (a diferencia de los requerimientos funcionales). Estos serán los requerimientos técnicos del sistema para que pueda cumplir sus funciones. En el siguiente listado se muestran los requerimientos propuestos para el software a

implementar, estos requerimientos al ser técnicos podrían estar sujetos a cambios dependiendo de dónde se vaya a implantar la solución. A continuación la lista:

Código	Descripción del Requerimiento
NFU-01	El sistema debe desarrollarse con herramientas de libre disponibilidad o que posean licencias gratuitas o académicas.
NFU-02	El explorador en el que se utilice la aplicación deberá tener soporte con mecanismos de geolocalización, tales como Chrome desde su versión 23.
NFU-03	El software debe ser desarrollado con estándares de programación y una arquitectura que permita que este sea escalable y mantenible en el tiempo.
NFU-04	La base de datos deberá tener la capacidad suficiente para almacenar por lo menos mil restaurantes.

2 Análisis de la solución

En la primera sección de este capítulo, se comenzó definiendo los requerimientos del sistema. En esta segunda parte, se detallará el análisis del sistema en base a los requisitos identificados. Para esto, se realizará un estudio técnico-económico, el catálogo de actores, los paquetes del sistema, los casos de uso, sus respectivas especificaciones, y el diagrama de clases.

2.1 Análisis Técnico y Económico

Para la elección de las herramientas se tuvo en consideración el tiempo de aprendizaje, el cual debe ser el mínimo para poder cumplir con el proyecto en el tiempo indicado. Asimismo, se ha tomado en cuenta utilizar solo aquellas que sean libres o aquellas con las que se cuente con alguna licencia de uso académico. Por otro lado, se contratarán servicios de hosting para poder subir la aplicación a un espacio de pruebas, estos serán los más sencillos que los proveedores ofrezcan por ser un software de carácter académico.

Considerando estos factores, se ha decidido por la utilización de tecnología .NET, en específico el lenguaje C# con el framework MVC3. La justificación es que se cuenta

con licencias académicas para poder utilizar estas herramientas. Asimismo, existe disponibilidad de web hosting para aplicaciones en .NET económicas que más adelante se detallarán en el análisis económico. Por otro lado, el motor de la base de datos será SQL Server, ya que se integra de forma eficiente con el lenguaje de programación debido a que pertenecen a la misma plataforma. Algunas ventajas de esta elección se muestran a continuación:

- Se puede trabajar en esta plataforma a través del IDE Visual Studio, el cual brinda muchas herramientas que permiten un **aprendizaje rápido** por parte de los desarrolladores.
- Su **Interoperabilidad Multilenguaje** permite soportar aplicaciones con componentes en múltiples lenguajes, lo cual hace posible poder trabajar todas las tecnologías involucradas (html, css, javascript, c#) dentro de un mismo entorno, haciendo así más eficiente el trabajo.
- La **documentación** de esta plataforma es amplia, lo cual es un apoyo para el desarrollador al elaborar sus tareas.

A continuación se presenta una estimación del costo que representa el desarrollo del proyecto de tesis, en el cual se cuenta con un total de 300 horas de trabajo:

Estimación del Proyecto de Tesis		
Costo por hora (S/.)	8.00	
Costos por Fase del Proyecto		
Fase	Horas	Costo (S/.)
Concepción Iteración 1	186	1488.00
Elaboración Iteración 1	48	384.00
Elaboración Iteración 2	124	992.00
Construcción Iteración 1	56	448.00
Construcción Iteración 2	200	1600.00
Total Trabajo	4912.00	

Total	4912.00
--------------	----------------

2.2 Catálogo de Actores

En este punto se indicará la clasificación de usuarios que interactuarán con el sistema basado en lo que realizará cada uno. Asimismo, en los anexos se presentará una breve descripción de cómo interactuarán con este y los resultados que esperan obtener. En la figura 2.1, se muestran los actores del sistema. De estos, los consumidores y usuarios públicos son roles similares, la diferencia se encuentra en que el usuario público está limitado en ciertas acciones dentro de la guía gastronómica (en los anexos se presenta una breve explicación de la función de cada actor).

▫

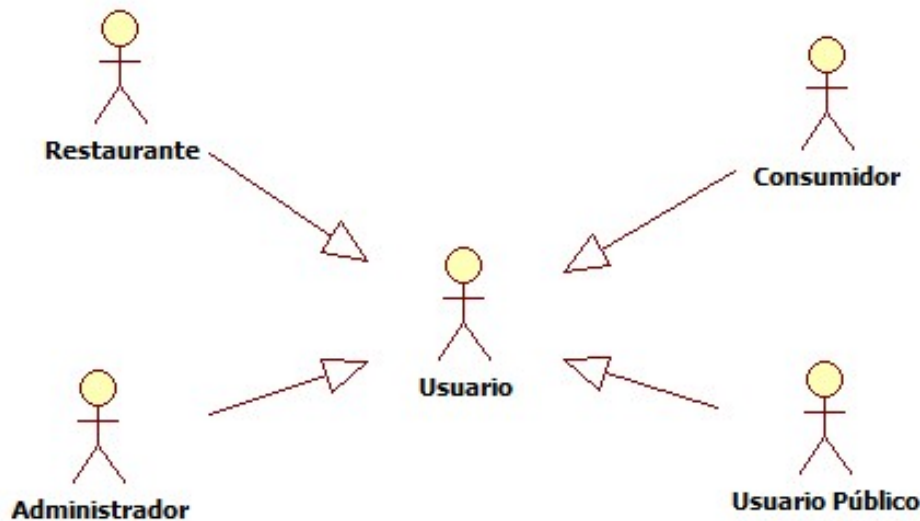


Figura 2.1 “Diagrama de Actores” (Elaboración Propia)

2.3 Paquetes

En esta sección, se muestra la paquetería que se aplicará al análisis del sistema con el fin de organizar el sistema para su mejor entendimiento. Estos paquetes englobarán casos de uso que cumplen con tareas estrechamente relacionadas.

- **Administración del Sistema:** englobará los casos de uso que se dediquen a la administración general del sistema, administración de los usuarios (restaurantes

y consumidores), administración de criterios de búsqueda y gestión de las críticas.

□



Figura 2.2 "Diagrama de Paquete" (Elaboración Propia)

- **Seguridad:** contendrá los casos de uso que se enfoquen en la seguridad del sistema como son los inicios de sesión, los cambios de contraseña y el desbloqueo de cuentas.
- **Flujo Restaurantes:** contendrá los casos de uso que se centren en la administración de los restaurantes por parte de los usuarios tipo restaurantes.
- **Flujo Consumidores:** englobará los casos de uso que se enfoquen en las acciones que pueden realizar los consumidores y los usuarios públicos.

2.4 Casos de Uso

Los casos de uso mostrarán las acciones que realizarán los actores del sistema. Estos estarán agrupados en los paquetes que se indicó en el punto anterior (el paquete de seguridad se encuentra en los anexos).

Paquete de Flujo Consumidores:

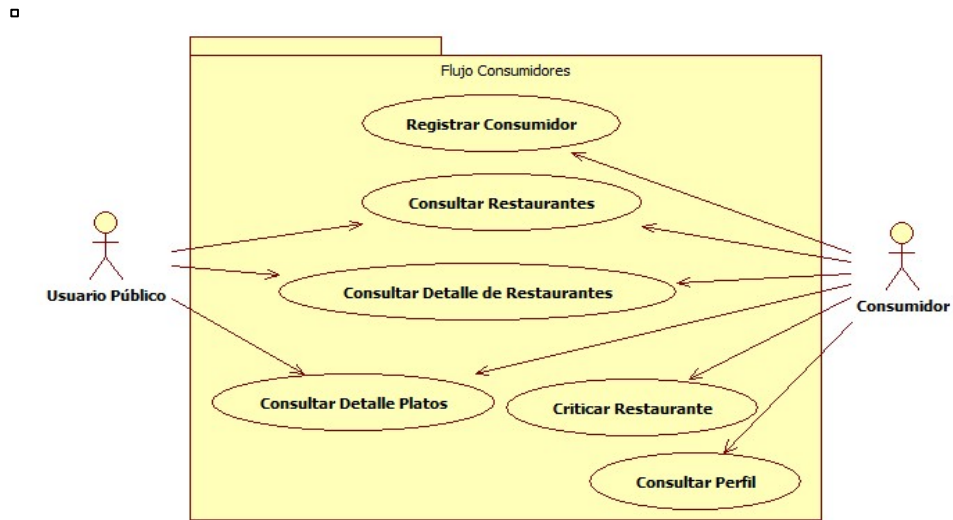


Figura 2.3 "Paquete Consumidores" (Elaboración Propia)

Paquete de Flujo Restaurantes:

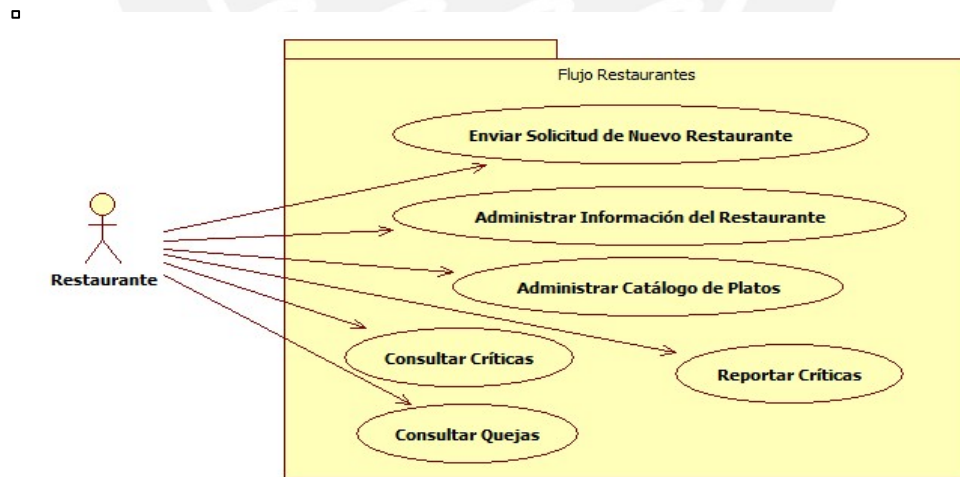


Figura 2.4 "Paquete Restaurantes" (Elaboración Propia)

Paquete de Administración del Sistema:

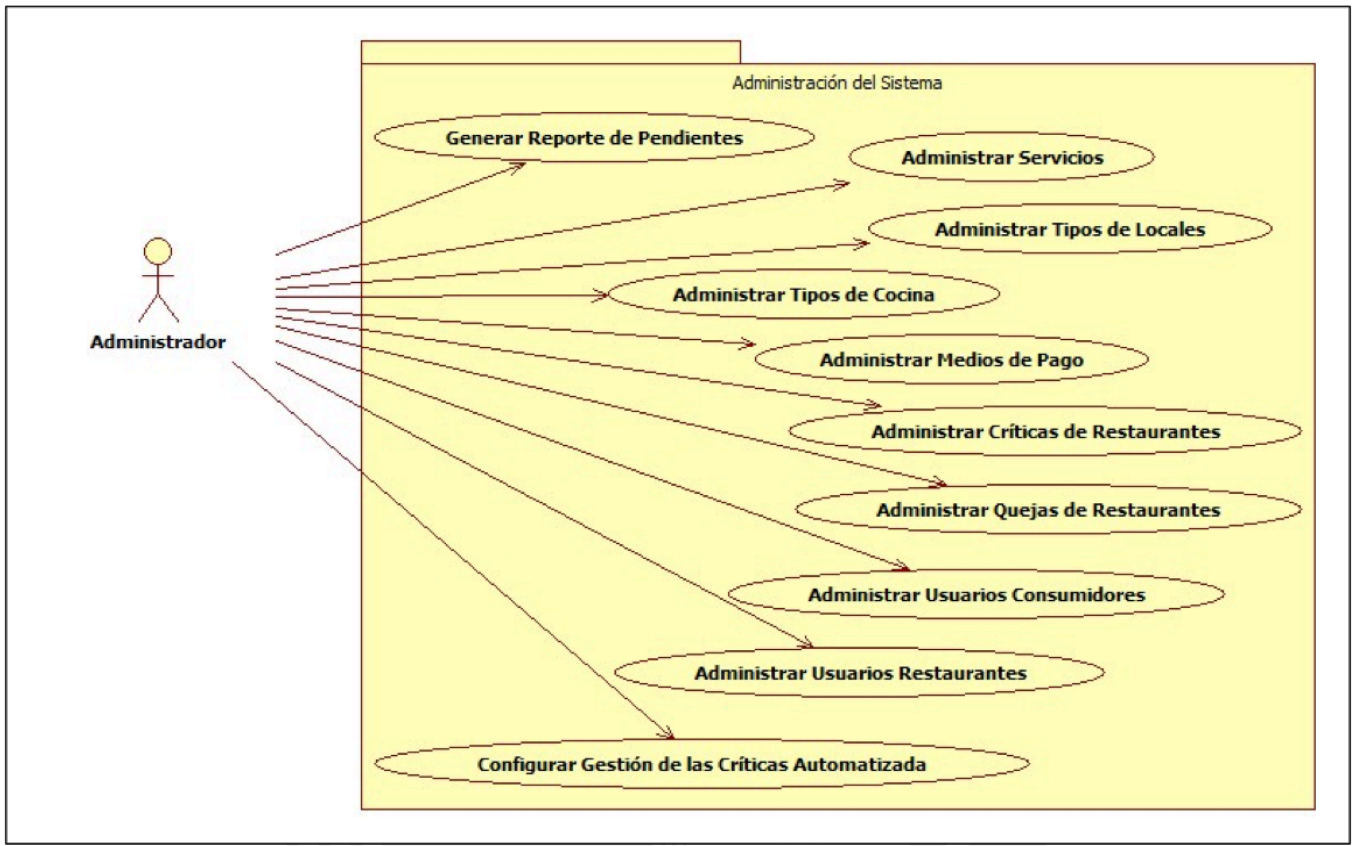


Figura 2.5 “Paquete Restaurantes” (Elaboración Propia)

2.5 Especificación de Casos de Uso

En este punto, se realizará la especificación de los casos de uso principales para poder entender el funcionamiento del sistema, estos también se presentarán organizados por paquetes como se hizo en las figuras 2.3, 2.4 y 2.5. En los anexos, se podrán encontrar más especificaciones de los casos de usos expuestos en el punto anterior.

Del paquete de “Flujo Consumidores”, los casos de uso más relevantes son los siguientes:

Consultar Restaurantes		
Datos	Id	CU-COM-002
	Descripción	Este caso de uso muestra cómo se realizan las búsquedas de restaurantes en el sistema.

Referencia a lista de requerimientos	FUN-32, FUN-33, FUN-34.
Paquete	Flujo Consumidores
Actores	Consumidor, Usuario Público.
Pre-condición	
	El actor del debe encontrarse en la página inicial sistema.
Flujo principal: “Búsqueda Normal”	
Acción del actor	Respuesta del Sistema
1. El actor debe ingresar el nombre de algún restaurante en el <i>Buscador</i> .	2. El sistema procesa la consulta y muestra un listado de restaurantes de acuerdo al filtro con los siguientes datos: <i>Nombre, Cocina, Distrito, Dirección, Calidad de Comida, Calidad de Servicio, Calidad de Ambiente, Precio Promedio, Cantidad de Votos y Distancia.</i>
Post-condición Flujo principal	
	Se mostró un listado de restaurantes.
Flujo alterno 1: “Búsqueda Avanzada”	
1. El actor selecciona la opción <i>Búsqueda Avanzada</i> .	2. El sistema muestra siete criterios distintos para la búsqueda: <i>Tipos de Cocina, Zona, Tipo de Local, Rango de Precios, Servicios, Medios de Pago y Distancia Máxima</i> . Cada criterio mostrará debajo suyo los valores a poder seleccionar para la búsqueda. Finalmente, se muestra dos opciones: <i>Buscar y Limpiar</i> .
3. El actor debe seleccionar todos los	4. El sistema mostrará el mismo

<p>critérios que desee según sus necesidades y seleccionar la opción <i>Buscar</i>.</p>	<p>resultado del punto 2 en el flujo principal.</p>
<p>Post-condición Flujo alterno 1:</p>	<p>Se mostró un listado de restaurantes.</p>

<p style="text-align: center;">Criticar Restaurante</p>		
<p>Datos</p>	<p>Id</p>	<p>CU-COM-005</p>
	<p>Descripción</p>	<p>Este caso de uso muestra el flujo de cómo el cliente califica a un restaurante.</p>
	<p>Referencia a lista de requerimientos</p>	<p>FUN-03, FUN-30, FUN-31.</p>
	<p>Paquete</p>	<p>Flujo Consumidores</p>
	<p>Actores</p>	<p>Consumidor.</p>
<p>Pre-condición</p>		<p>El actor debe haber iniciado sesión.</p>
<p>Flujo principal: “Calificar Restaurante”</p>		
<p>Acción del actor</p>		<p>Respuesta del Sistema</p>
<p>1. El actor consulta el detalle de un restaurante siguiendo el flujo del caso de uso CU-COM-003.</p>		<p>2. El sistema mostrará el detalle de todo el restaurante.</p>
<p>3. El actor selecciona la opción <i>Calificar Restaurante</i>.</p>		<p>4. El sistema mostrará el <i>nombre del restaurante</i>. Asimismo, mostrará cuatro criterios de evaluación: <i>Comida, Servicio, Ambiente y Precio</i>. Los tres primeros criterios podrán ser calificados bajo el siguiente rango: <i>Mala, Regular, Buena, Muy Buena, Excelente</i> que corresponden a los valores de 1 a 5. En cuanto al criterio de <i>Precio</i></p>

	se ingresará un precio promedio que incluya entrada, plato principal y bebidas. Finalmente, se mostrará un campo que se llame <i>Comentarios</i> , en el cual se ingresará un texto, y la opción <i>Calificar</i> .
5. El actor llenará los campos mencionados en el punto anterior y seleccionará la opción <i>Calificar</i> .	6. El sistema mostrará el siguiente mensaje: <i>Gracias por calificar al restaurante X, se procesará su comentario lo más rápido posible para que sea publicado en la página web.</i>
Post-condición Flujo principal	Se registró una calificación en el sistema.
Flujo alternativo 1: “Criterios Vacíos”	
1. En el punto 5 del flujo principal, el actor no llena todos los criterios de calificación y selecciona la opción <i>Calificar</i> .	2. El sistema valida los criterios y, para los que estén vacíos, mostrará el mensaje <i>Por favor indique un valor para este criterio.</i>
3. El actor volverá al punto 5 del flujo principal hasta que llene correctamente todos los criterios y seleccione la opción <i>Calificar</i> .	4. El sistema mostrará el siguiente mensaje: <i>Gracias por calificar al restaurante X, se procesará su comentario lo más rápido posible para que sea publicado en la página web.</i>
Post-condición Flujo alternativo 1:	Se registró una calificación en el sistema.
Flujo alternativo 2: “Comentarios incompletos”	
1. En el punto 5 del flujo principal, el actor llena los criterios, coloca un comentario con menos de 100 caracteres y selecciona la opción	2. El sistema valida los criterios y los comentarios, y muestra un mensaje <i>La reseña debe tener más de 100 caracteres.</i>

<i>Votar.</i>	
3. El actor volverá al punto 5 del flujo principal hasta que llene correctamente los criterios y el comentario y seleccione la opción <i>Calificar</i> .	4. El sistema mostrará el siguiente mensaje: <i>Gracias por calificar al restaurante X, se procesará su comentario lo más rápido posible para que sea publicado en la página web.</i>
Post-condición Flujo alterno 2:	Se registró una calificación en el sistema.

Del paquete de “Administración del Sistema”, los casos de uso más relevantes son los siguientes:

Administrar Críticas de Restaurantes		
Datos	Id	CU-ADMIN-006
	Descripción	Consiste en administrar las críticas realizadas por los consumidores hacia los restaurantes.
	Referencia a lista de requerimientos	FUN-10, FUN-11.
	Paquete	Administración del Sistema
	Actores	Administrador
Pre-condición	El actor debe haber iniciado sesión.	
Flujo principal: “Acepta Críticas”		
Acción del actor	Respuesta del Sistema	
1. El actor debe seleccionar la opción <i>Críticas</i> .	2. El sistema mostrará un listado con el <i>código de la crítica</i> , el <i>nombre del comensal</i> , el <i>nombre del restaurante</i> , la <i>fecha</i> en que se registró, su <i>estado</i> , el cual será <i>Pendiente</i> para los que aún no han sido revisados, y la opción <i>Editar</i> .	

<p>3. El actor seleccionará la opción <i>Editar</i> correspondiente a cualquiera de las críticas.</p>	<p>4. El sistema mostrará un listado de las críticas con los siguientes datos: <i>código de la crítica, nombre del usuario, fecha de registro y estado</i>. Este último será el único editable para poder aceptar o rechazar la crítica. Por último, se muestra la opción <i>Guardar cambios</i>.</p>
<p>5. El actor asignará el estado <i>Activado</i> y seleccionará la opción <i>Guardar cambios</i>.</p>	<p>6. El sistema muestra el siguiente mensaje: <i>La crítica se ha activado en el sistema</i>.</p>
<p>Post-condición Flujo principal</p>	<p>La crítica se encuentra en estado activado en el sistema.</p>
<p>Flujo alternativo 1: “Rechazar Críticas”</p>	
<p>1. En el punto 5 del flujo principal, el actor asigna el estado <i>Desactivado</i>.</p>	<p>2. El sistema muestra un cuadro con el título de <i>Razones</i>, en el cual se redacta las razones por las que fue rechazada la crítica. Además, se muestra la opción <i>Enviar</i> y <i>Cancelar</i>.</p>
<p>3. El actor redacta las razones correspondientes en el cuadro y selecciona la opción <i>Enviar</i>.</p>	<p>4. El sistema muestra el siguiente mensaje: <i>Se ha enviado un correo al comensal indicándole las razones por las que no pudo registrarse su crítica en el sistema</i>. Además, se muestra la opción <i>Aceptar</i>.</p>
<p>5. El actor selecciona la opción <i>Aceptar</i></p>	<p>6. Se muestra el listado inicial del punto 2 del flujo principal.</p>
<p>Post-condición Flujo alternativo 1:</p>	<p>Las críticas aceptadas quedarán registradas en el sistema.</p>

Configurar Gestión de las Críticas Automatizada		
Datos	Id	CU-ADMIN-010
	Descripción	Consiste en activar y desactivar el mecanismo que administra las críticas automáticamente, así como la administración de las palabras que usará como patrones al analizar los textos.
	Referencia a lista de requerimientos	FUN-16, FUN-17.
	Paquete	Administración del Sistema
	Actores	Administrador
Pre-condición		El actor debe haber iniciado sesión.
Flujo principal: “Activar gestión automatizada”		
Acción del actor		Respuesta del Sistema
1. El actor debe seleccionar la opción <i>Configuración</i> .		2. El sistema mostrará una sección llamada <i>Gestión de las críticas</i> , en esta se mostrarán las siguientes opciones: <i>Registrar nueva palabra</i> y <i>Activar gestión automatizada</i> . Asimismo, se mostrará un listado de palabras, en el cual se mostrará su <i>código</i> , la <i>palabra</i> , y las opciones <i>Editar</i> y <i>Eliminar</i> .
3. El actor selecciona la opción <i>Activar gestión automatizada</i> .		4. El sistema mostrará un mensaje de verificación indicando lo siguiente: <i>¿Desea realmente activar la gestión automatizada?</i> Asimismo, se mostrarán las opciones <i>Sí</i> y <i>No</i> .
5. El actor seleccionará la opción <i>Sí</i> .		6. El sistema mostrará un mensaje indicando cuántas críticas fueron aceptadas y rechazadas.
Post-condición Flujo principal		El mecanismo de administración de

	críticas automatizado se encuentra activado en el sistema.
Flujo alternativo 1: “Registrar palabra”	
7. En el punto 3 del flujo principal, selecciona la opción <i>Registrar nueva palabra</i> .	8. El sistema muestra una campo editable en el que se ingresará la <i>palabra</i> a registrar y la opción <i>Registrar</i> .
9. El actor redacta la nueva palabra y seleccionará la opción <i>Registrar</i> .	10. El sistema muestra el siguiente mensaje: <i>Se ha registrado la palabra exitosamente</i> . Además, muestra la opción <i>Aceptar</i> .
11. El actor selecciona la opción <i>Aceptar</i> .	12. Se muestra el listado inicial del punto 2 del flujo principal.
Post-condición Flujo alternativo 1:	La palabra ha quedado registrada en el sistema.

Por último, del paquete de “Flujo Restaurantes”, los casos de uso más relevantes son los siguientes:

Administrar Información del Restaurante		
Datos	Id	CU-REST-002
	Descripción	Consiste en que el usuario restaurante pueda administrar la información que publicará en el sistema.
	Referencia a lista de requerimientos	FUN-02, FUN-19, FUN-20, FUN-21, FUN-22, FUN-23.
	Paquete	Flujo Restaurantes
	Actores	Restaurante
Pre-condición		El actor ha iniciado sesión.
Flujo principal: “Administrar Información”		

Acción del actor	Respuesta del Sistema
1. El actor debe seleccionar la opción <i>Administrar información</i> .	2. El sistema mostrará todos los datos del restaurante registrados durante el paso 2 del caso de uso CU-REST-001 y mostrará la opción <i>Guardar cambios</i> .
3. El actor podrá editar todos los datos generales del restaurante. Asimismo, podrá agregar o quitar <i>medios de pago y servicios que brinda</i> . Además, podrá cambiar la ubicación del restaurante en el mapa, así como cambiar la imagen del local. Finalmente, seleccionará la opción <i>Guardar cambios</i> .	4. El sistema mostrará el siguiente mensaje: <i>La información ha sido actualizada</i> . Y se presentará la opción <i>Aceptar</i> .
5. El actor seleccionará la opción <i>Aceptar</i> .	6. El sistema redireccionará hacia la página principal del usuario restaurante.
Post-condición Flujo principal	Se actualizaron los datos del restaurante en el sistema.

2.6 Diagrama de Clases

En la figura 2.6, se muestra el Diagrama de Clases correspondiente al sistema, el cual sigue la sintaxis dada por UML para modelar el software.

En este diagrama, se puede observar que hay una clase aislada denominada “Opcion”. La utilidad de esta se encuentra en que lleva un registro de las palabras que servirán de patrones para realizar el control del contenido textual de las críticas. En el capítulo 3 de diseño, se le volverá a visualizar en el diagrama de base de datos.

□

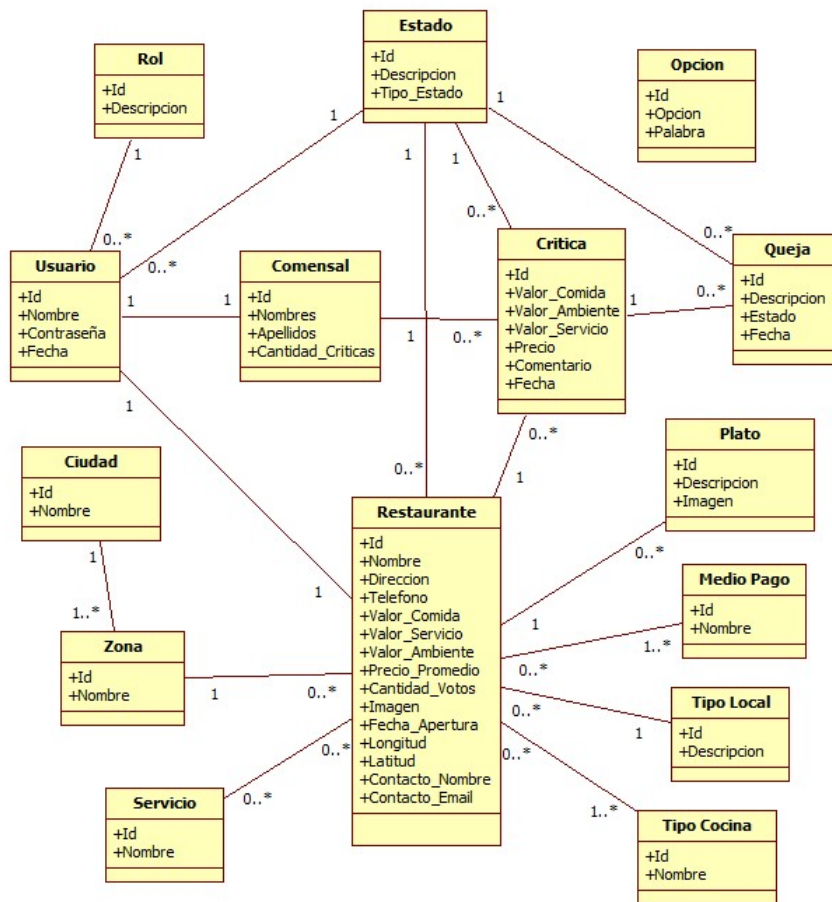


Figura 2.6 "Diagrama de Clases" (Elaboración Propia)

3 Arquitectura de la Solución

En este capítulo, se explicará la arquitectura construida para poder brindarle soporte a la solución que este proyecto presenta, mostrando así todos sus componentes, de qué se encarga cada uno y cómo interactúan entre ellos.

En la presente sección, se presentan los diferentes componentes del sistema, la colaboración de cada uno de ellos y cómo interactúan. Para esto, se utilizan ciertos principios del modelo de "4+1 Vistas" de la metodología RUP. Para la descripción de la arquitectura, se utilizarán las vistas lógicas, de implementación y de despliegue, ya que así se podrá visualizar el sistema en capas abstractas, reconocer los componentes internos de cada una de estas y los componentes físicos (hardware) que serán necesarios

para su funcionamiento. Asimismo, para la elaboración de esta arquitectura se ha repasado los requerimientos que se especificaron en el capítulo 2 de Análisis para poder cumplir con ellos.

3.1 Vista Lógica

La vista lógica permite visualizar el sistema en capas abstractas. La estructura que se presenta aquí, es la forma en que se plantea resolver los requerimientos del sistema. Para esto, se engloban a las clases del sistema en capas, las cuales tienen distintos objetivos y formas de interactuar entre ellas. Así, esta vista permite al desarrollador y a los clientes tener una idea clara de cómo estará organizado el sistema a desarrollar desde un punto de vista de alto nivel.

En la figura 2.7, se muestra la vista lógica. En esta lo que se quiere presentar es que la arquitectura estará compuesta por tres capas principalmente: una de presentación o interfaz gráfica, otra con la lógica de la aplicación y la última con el manejo de los recursos. A continuación se detalla el comportamiento de cada capa:

- **Modelo**

En esta capa, se representa la información sobre la cual el sistema operará. Para este sistema, recordando la figura 2.6 "Diagrama de Clases", las entidades como Restaurantes, Quejas, Comentarios, Usuarios, Solicitudes, entre otros, serían accedidas por esta capa para poder manejar su información.

- **Vista**

Esta capa se encarga de ilustrar los elementos de la capa de modelo en formas adecuadas para que el usuario pueda interactuar con la información del sistema. En este sistema, lo usual será renderizar páginas HTML con información cargada dinámicamente por código.

- **Controlador**

Esta capa define el comportamiento del sistema. Esta se encarga de procesar y responder a eventos que los usuarios realizan, por lo que se encarga de modificar los modelos y vistas.

Siguiendo la estructura de la figura 2.7, el sistema se comportará de la siguiente forma:

1. Los usuarios podrán interactuar con este a través de las interfaces.
2. Los controladores manejarán esos eventos que provienen de las interfaces.
3. Los controladores accederán al modelo y lo modificarán dependiendo de la acción que el usuario haya realizado.
4. Los modelos notificarán a las vistas los cambios que se hayan realizado, y también podrán ser consultados por las vistas directamente.
5. La interfaz espera hasta que el usuario vuelva a realizar un evento sobre este.

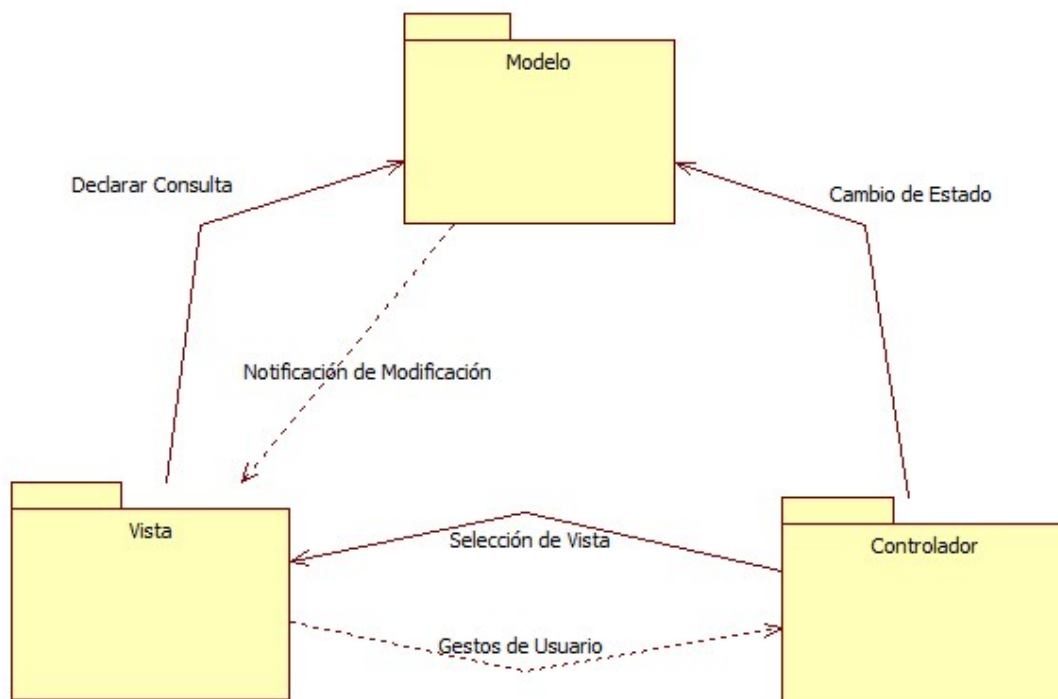


Figura 2.7 "Arquitectura - Vista Lógica" (Elaboración Propia)

3.2 Vista de Implementación

En esta vista, se podrá visualizar los componentes internos de las capas definidas en la vista lógica. A diferencia de la vista lógica, esta se enfoca más en el detalle de los componentes del sistema y cómo interactúan entre ellos, así como las interfaces que uno le brinda a otro.

En la figura 2.8, se puede visualizar los componentes por cada capa. A continuación se presenta una descripción de ellos:

- **Componentes GUI Estáticos**

Estos son el conjunto de componentes que se encargan de redenzar las interfaces a los usuarios de forma estática. Como el entorno es web, lo que se utiliza para este caso es HTML 5 para mostrar el contenido y CSS 3 para el formato.

- **Componentes GUI Dinámicos**

Estos son el conjunto de componentes que se encargan de brindar el contenido dinámico a los componentes estáticos. En este caso, se utilizará AJAX para poder realizar llamadas asíncronas al servidor y no necesitar refrescar las interfaces en algunos casos. Asimismo, debido a que se utilizará el framework MVC3 de .NET, se utilizará el motor gráfico Razor que genera archivos CSHTML para poder brindar el contenido dinámico que se obtiene del servidor.

- **Visualización de Mapas**

Para la visualización de los mapas se utilizará la librería de google: Google Maps Web API versión 3. Esta permitirá a los usuarios registrar los restaurantes con una ubicación en el mapa, visualizar a los restaurantes dentro de un mapa, y encontrar las distancias con respecto al usuario a través de su mecanismo de geolocalización que es soportado por los exploradores indicados en los requerimientos no funcionales del capítulo 2 de análisis.

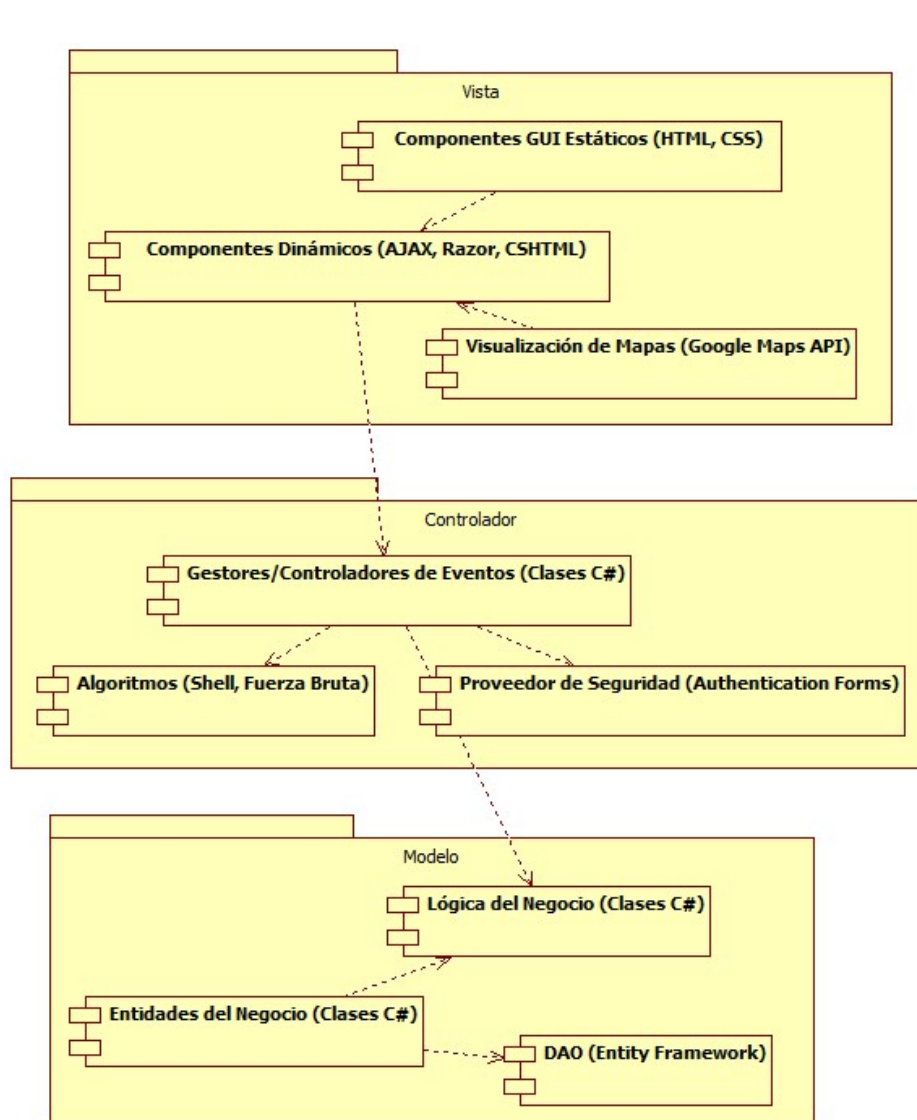


Figura 2.8 "Diagrama de Componentes" (Elaboración Propia)

- **Visualización de Mapas**

Para la visualización de los mapas se utilizará la librería de google: Google Maps Web API versión 3. Esta permitirá a los usuarios registrar los restaurantes con una ubicación en el mapa, visualizar a los restaurantes dentro de un mapa, y encontrar las distancias con respecto al usuario a través de su mecanismo de geolocalización que es soportado por los exploradores indicados en los requerimientos no funcionales del capítulo 2 de análisis.

- **Gestores/Controladores de Eventos**

Para que las acciones realizadas sobre la interfaz gráfica por parte de los usuarios finales tengan efecto sobre la data del sistema, se utilizarán los gestores o controladores de eventos que son parte del framework de MVC3. Estos captarán los eventos y procesarán la data necesaria. Una vez hecho este procesamiento, el controlador se comunica una vez más con la interfaz para devolver la vista (el resultado). Para realizar esto, los controladores necesitarán hacer uso de los proveedores de seguridad para validar y verificar los roles de los usuarios; y también podrían necesitar hacer uso de los algoritmos dependiendo del evento captado.

- **Proveedor de Seguridad**

Para la autenticación de los usuarios y sus roles es necesario proveer mecanismos de seguridad. En este caso, se utilizarán los Authentication Forms de .NET, la cual es una librería que contiene las funcionalidades básicas y más importantes de los mecanismos de seguridad. Asimismo, esta puede extenderse para adaptarse a las necesidades de la aplicación e integrarse con la base de datos en uso.

- **Algoritmos**

En este caso los controladores podrán necesitar interactuar con los algoritmos expuestos en los objetivos específicos del capítulo 1 y expuestos nuevamente en el capítulo 2 de análisis. Esto se debería a los casos en los que se de un ordenamiento de las distancias de los restaurantes con respecto al usuarios y en el caso de un evento que necesite el control del contenido textual de los comentarios.

- **Lógica del Negocio**

En este componente se tiene a todas la clases en C# que moldearán el comportamiento del software. En este caso, este será el puente que interactuará mayormente con las Entidades del Negocio para poder consultar la data del sistema y

luego poder brindársela a los Componentes Dinámicos y finalmente poder renderizar el contenido en un portal web.

- **Entidades del Negocio**

En este componente se tiene a todas las clases en C# que permiten interactuar con la data del sistema. En este componente se encontrarán todas aquellas funcionalidades que permiten tanto consultar data como insertarla en la base de datos, por lo que siempre será convocado por la Lógica de Negocio cuando sea necesario. Asimismo, para que este componente pueda acceder a la base de datos, depende de otro componente, el DAO, el cual brinda los mecanismos necesarios para poder acceder a ella.

- **DAO**

Este último componente, brinda los mecanismos necesarios para poder interactuar con la base de datos sin necesidad de usar SQL directamente. A través del Entity Framework, esto es posible por el mapeo que se realiza entre las clases del sistema y las tablas de la base de datos.

3.3 Vista de Despliegue

En esta vista, se podrá visualizar finalmente como el sistema explicado en las vistas anteriores se mapea con el hardware, es decir, los dispositivos físicos que se utilizarán en última instancia.

En la figura 2.9, se puede visualizar los principales nodos en los que se distribuirá el sistema. A continuación se hace una explicación de cada uno:

- **Servidor Web y de Aplicación**

Este nodo representa al servidor que contendrá toda la lógica de la aplicación, es decir, mantendrá en ejecución constante los compilados del código fuente. En la figura,

se muestran los componentes que estarán inmersos en este nodo.

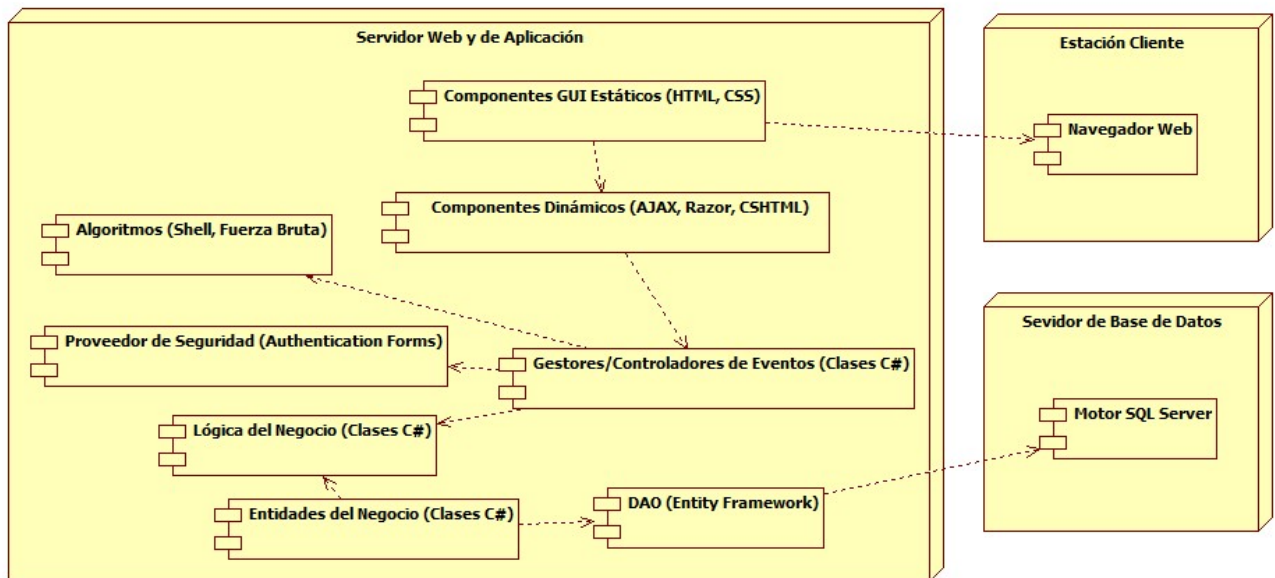


Figura 2.9 “Diagrama de Despliegue” (Elaboración Propia)

- **Estación Cliente**

Este nodo representa a cualquier estación con navegadores web y acceso a internet. En este todos los usuarios podrán ingresar al sistema por medio del navegador e interactuar con él.

- **Servidor de Base de Datos**

Este nodo consistirá en otro servidor que contenga al motor de base de datos. Este servidor podrá interactuar con el Servidor Web y de Aplicación por medio del DAO.

CAPÍTULO 3: ORDENAMIENTO DE RESTAURANTES

Este capítulo se centra en desarrollar y resolver el segundo objetivo específico, el cual resulta en un mecanismo que permita mostrar los restaurantes de forma ordenada en base a la distancia con respecto a los usuarios. Para esto se empezará mencionando algunos de los elementos del sistema con los que el mecanismo se integrará, así como el uso que se hizo de las herramientas necesarias para su funcionamiento. Finalmente, se presentará el algoritmo y su adaptación al sistema.

1 Integración con el sistema

En el capítulo de análisis y arquitectura, se mostraron todos los elementos del sistema. En esta sección, se señalarán solo aquellos elementos del sistema con los que interactúa el mecanismo y cómo.

El mecanismo es requerido dentro de una de las funcionalidades enfocadas en el comensal, esta es la funcionalidad 33 *“El sistema debe permitir mostrar los resultados de las búsquedas con la distancia entre el usuario y los locales, así como permitir ordenarlos en base a dicha distancia”*. Esta funcionalidad se agrupó dentro del caso de uso CU-COM-002 *“Consultar Restaurantes”*, cuya postcondición es mostrar un listado de restaurantes. Los resultados de este listado serán ordenados en base al algoritmo. En la figura 3.1, se muestra el paquete en el que se agrupa este mecanismo.



Figura 3.1 "Paquete con mecanismo de ordenamiento" (Elaboración Propia)

Por otro lado, los datos del sistema que serán utilizados por el algoritmo serán los de la posición de cada restaurante. Siguiendo con el diagrama de clases, estos datos serían los de Longitud y Latitud que, a su vez, se reflejan en el diagrama de base de datos. En la figura x.x, se muestra la clase relevante para este mecanismo.

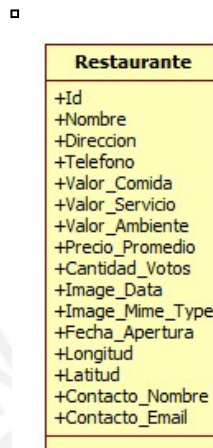


Figura 3.2 “Clases: mecanismo de ordenamiento” (Elaboración Propia)

2 Prerequisitos del algoritmo

En el capítulo 1, en la sección de métodos, se mencionaron algunas de las herramientas que se utilizarían en el desarrollo de cada mecanismo. En esta sección, se muestra cómo se utilizaron dichas herramientas para la obtención de dos datos importantes para el desarrollo del algoritmo: la posición del usuario y el cálculo de distancias entre este y los restaurantes.

2.1 Geolocalización en Navegadores de Internet

Uno de los elementos necesarios para poder obtener las distancias (que luego serán comparadas y ordenadas por el algoritmo), es la ubicación de los usuarios que están realizando la búsquedas, es decir, los comensales.

Debido a que la plataforma es web, es necesario obtener las coordenadas de posición de los usuarios a través de los navegadores de internet. Esto es posible en la mayoría de navegadores en sus últimas versiones gracias a la geolocalización.

La geolocalización permite hallar dónde un usuario se encuentra en el mundo. Existe más de una forma de conseguir esto: a través de la dirección IP, la conexión red inalámbrica con la que se trabaja, la antena de telefonía móvil (en el caso que se ingrese desde un celular) o el hardware de GPS. Esta funcionalidad es soportada en exploradores tales como IE, versión 9 hacia arriba; Firefox, versión 10 hacia arriba; Safari, versión 5 hacia arriba; Chrome, versión 5 hacia arriba; entre otros. (W3C 2015)

La funcionalidad de la geolocalización es accesible a través de javascript. Esta se encuentra en una nueva propiedad del objeto global “navigator”: navigator.geolocation. La forma en que se ha utilizado en el proyecto se visualiza en la figura 3.3, en la que la posición (latitud y longitud) es obtenida en las primeras líneas a través del método “getCurrentPosition” que es parte del objeto “navigator.geolocation”. Este método permite tener acceso a las coordenadas a través de su atributo “coords”. Esta figura es parte de un archivo cshtml del cual se hablará más en el capítulo de construcción.

```

222     navigator.geolocation.getCurrentPosition(function (position) {
223         var pos = new google.maps.LatLng(position.coords.latitude,
224                                         position.coords.longitude);
225         var variables = "lng="+pos.lat()+"&lat="+pos.lng();
226         $("#listRestaurantes").attr("href", "@Url.Action("Index", "Restaurante")?" + variables);
227     }, function () {
228         handleNoGeolocation(true);
229     });
230
231     function handleNoGeolocation(errorFlag) {
232         if (errorFlag) {
233             var content = 'Error: The Geolocation service failed.';
234         } else {
235             var content = 'Error: Your browser doesn\'t support geolocation.';
236         }
237     }
238 }
  
```

Figura 3.3 “Uso de geolocalización” (Elaboración Propia)

2.2 Fórmula de Haversine

En el punto anterior, se mencionó cómo hallar la posición (latitud y longitud) del usuario para poder luego calcular la distancia entre este y los restaurantes. Dicha distancia tiene un cálculo especial, ya que estamos tomando en cuenta dos puntos en la superficie de la Tierra. La fórmula del Haversine permite calcular distancias para este tipo de casos.

La fórmula del Haversine permite calcular la distancia de grandes círculos entre dos puntos sobre una esfera a partir de sus latitudes y longitudes, esto es, la distancia más corta sobre la superficie de la Tierra. Dicha fórmula tiene sus propias limitaciones

como el hecho de ignorar los accidentes de la misma Tierra: montañas, ríos, entre otros (GOODWIN 1910). La fórmula se puede visualizar en la figura x.x, en la que la diferencia de phi es la diferencia de latitudes, la diferencia de lambda es la diferencia de longitudes y R es el radio de la Tierra, es decir, 6731km.

$$\begin{aligned}
 a &= \sin^2(\Delta\phi/2) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2(\Delta\lambda/2) \\
 c &= 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \\
 d &= R \cdot c
 \end{aligned}$$

Figura 3.4 “Fórmula de Haversine” (GOODWIN 1910)

El código implementado en C# de la fórmula se puede apreciar en la figura 3.5. La implementación y la fórmula no ha variado mucho, ya que la versión de la fórmula que se presentó estaba enfocada en el cálculo vía computadora. En la implementación, se puede apreciar que la diferencia de latitudes y longitudes se convirtieron en radianes a través de la fórmula “Radio”, esta conversión es necesaria antes de pasar la diferencia a las fórmulas trigonométricas.

```

public static double Radio (double distancia)
{
    return distancia*System.Math.PI/180;
}

//KM
public static double Distance(double latitud1, double longitud1, double latitud2, double longitud2)
{
    var R = 6371; // earth's mean radius in km
    var dLat = Radio(latitud2 - latitud1);
    var dLong = Radio(longitud2 - longitud1);

    var a = System.Math.Sin(dLat/2) * System.Math.Sin(dLat/2) +
            System.Math.Cos(Radio(latitud1)) * System.Math.Cos(Radio(latitud2)) *
            System.Math.Sin(dLong/2) * System.Math.Sin(dLong/2);
    var c = 2 * System.Math.Atan2(System.Math.Sqrt(a), System.Math.Sqrt(1-a));
    var d = R * c;

    return d;
}
  
```

Figura 3.5 “Implementación Fórmula de Haversine” (Elaboración Propia)

2.3 Google Maps API Web Services

En el punto anterior, se mencionó que la fórmula de Haversine nos permite hallar la distancia entre dos puntos en la Tierra, con ciertas limitaciones, pero bastante aproximado. Asimismo, la implementación de dicha fórmula significa que el cálculo de

la misma se realizaría en el servidor web que aloje la plataforma, por lo que dependerá de la potencia del mismo su desempeño. Ya que es probable que el cálculo de estas distancias puedan resultar en una sobre carga del servidor (y se le quiera de forma más precisa: considerando calles, etc), se tomó en cuenta una alternativa que permita desacoplar dicho cálculo. Para esto se utilizó los Google Maps API Web Services (GMWS).

Los GMWS son una colección de interfaces HTTP a los servicios de Google que proveen data geográfica para aplicaciones con mapas. De los servicios que provee, el que nos permite calcular las distancias, en base a las coordenadas, es el “Distance Matrix API”. A diferencia de la fórmula de Haversine por sí sola, este servicio se basa en la ruta recomendada entre un punto de inicio y uno de llegada para calcular una distancia. Estas rutas solo son posibles de considerar en servicios como los de Google Maps, en el cual calles, avenidas, entre otros elementos urbanos, son tomados en cuenta. Para consultar rutas a este servicio, se realiza una llamada http como se muestra en la figura 3.6, en la cual los parámetros requeridos son una lista de coordenadas orígenes y coordenadas destino. Para este caso de la distancia, las coordenadas de origen son siempre la posición del usuario, por lo que se mantienen iguales las coordenadas de origen.

```

Dirección de consulta:
https://maps.googleapis.com/maps/api/distancematrix/output?parameters

parameters:
origins=41.43206,-81.38992|41.43206,-81.38992|41.43206,-81.38992
destinations=42.43206,-83.38992|44.43206,-80.38992|45.43206,-86.38992
  
```

Figura 3.6 “Consulta a Servicios de Google” (Elaboración Propia)

Finalmente, las distancias son devueltas dentro de un JSON como se muestra en la figura 3.7, a las cuales se les transforma en un tipo de dato adecuado para poder ser ordenadas por el algoritmos Shell.

```

{
  "status": "OK",
  "origin_addresses": [ "direccion_origen_n" ],
  "destination_addresses": [ "direccion_destino_n" ],
  "rows": [ {
    "elements": [ {
      "status": "OK",
      "duration": {
        "value": 340110,
        "text": "3 hours 22 minutes"
      },
      "distance": {
        "value": 1734542,
        "text": "1 735 km"
      }
    }, {
      "status": "OK",
      "duration": {
        "value": 24487,
        "text": "6 horas 48 minutes"
      },
      "distance": {
        "value": 129324,
        "text": "129 km"
      }
    }
  ]
} ]
}

```

Figura 3.7 “Respuesta de Servicio de Google” (Elaboración Propia)

Si bien esta herramienta resulta mejor en términos de precisión en cuanto a las distancias, también tiene sus limitaciones en su versión gratuita, su uso no puede exceder los 100 elementos por consulta, 100 por 10 segundos o 2500 por un periodo de 24 horas. Por lo que se tendría que analizar si el costo de esta herramienta es aceptable para los beneficios que da frente al uso de la fórmula de Haversine.

3 Algoritmos de Ordenamiento “Shell”

Con lo desarrollado en la sección anterior, ya se puede empezar a analizar e implementar el algoritmo de ordenamiento, dado que ahora se cuenta con el cálculo de las distancias.

El algoritmo de ordenamiento Shell es el que se utilizará para el ordenamiento mencionado. Al final de esta descripción, se puede visualizar el pseudocódigo del algoritmo adaptado para la solución que se está presentando. Para su adaptación, se tomó como base el diagrama de clases mostrado en el capítulo 2 y las referencias del algoritmo.

En el algoritmo presentado, el arreglo a ordenar serían las distancias de los restaurantes consultados con respecto al usuario. En cuanto al algoritmo en sí, este consiste en un primer bucle que hace posible los “saltos”, los cuales generarán los sub-arreglos en los siguientes bucles anidados. Estos sub-arreglos son los que son ordenados de acuerdo a la cantidad de saltos que se creen.

```

#Ordenar un arreglo a[0..n-1]

k = n/2
while (k >= 1){
  for (i = k; i < n; i++){
    v = a[i]
    j = i - k
    while (j >= 0 and a[j] > v){
      a[j + k] = a[j]
      j = j - k
    }
    a[j + k] = v
  }
  k = k/2
}

```

En la figura 3.8, se muestra la implementación del algoritmo para el cual se han utilizado “listas” como el tipo de dato para en reemplazo del arreglo que se muestra en el pseudocódigo.

```
65 public static void ShellSort(List<DistanciaRest> listaDistancias)
66 {
67     //List<DistanciaRest> listaDistanciasOrdenadas = new List<DistanciaRest>();
68
69     var n = listaDistancias.Count();
70     var k = n/2;
71
72     while (k >= 1)
73     {
74         for (int i=k; i<n;i++)
75         {
76             var v = listaDistancias.ElementAt(i);
77             var j = i-k;
78             while (j>=0 && listaDistancias.ElementAt(j).Distancia > v.Distancia)
79             {
80                 listaDistancias.Insert(j+k, listaDistancias.ElementAt(j));
81                 listaDistancias.RemoveAt(j + k +1);
82                 j = j - k;
83             }
84             listaDistancias.Insert(j+k,v);
85             listaDistancias.RemoveAt(j + k + 1);
86         }
87         k = k / 2;
88     }
89 }
```

Figura 3.8 “Implementación Algoritmo Shell” (Elaboración Propia)



CAPÍTULO 4: ANÁLISIS DE COMENTARIOS

Este capítulo se centra en desarrollar y resolver el tercer objetivo específico, el cual resulta en un artefacto que permita automatizar el análisis de los comentarios registrados. Para esto primero se explicará cómo este se integra con el sistema y luego se explicará y mostrará la implementación del algoritmo que lo soporta.

1 Integración con el sistema

Similar a la primera sección del capítulo anterior, en esta se señalarán solo aquellos elementos del sistema con los que interactúa el mecanismo y cómo.

El mecanismo es requerido dentro de una de las funcionalidades enfocadas en la administración del sistema, esta es la funcionalidad 11 *El sistema debe permitir que las críticas puedan ser aceptadas o rechazadas automáticamente a través de un algoritmos que identifique texto con palabras malintencionadas*. Asimismo, el mecanismo está relacionado con las funcionalidades 16 y 17, ya que estas especifican algunas de sus fuentes para su funcionamiento. Estas funcionalidades se agruparon dentro de los casos de uso CU-ADMIN-006 y CU-ADMIN-010 (*Administrar Críticas de Restaurantes y Configurar Gestión de las Críticas Automatizadas*). En la figura 4.1, se muestra el paquete en el que se agrupa este mecanismo.

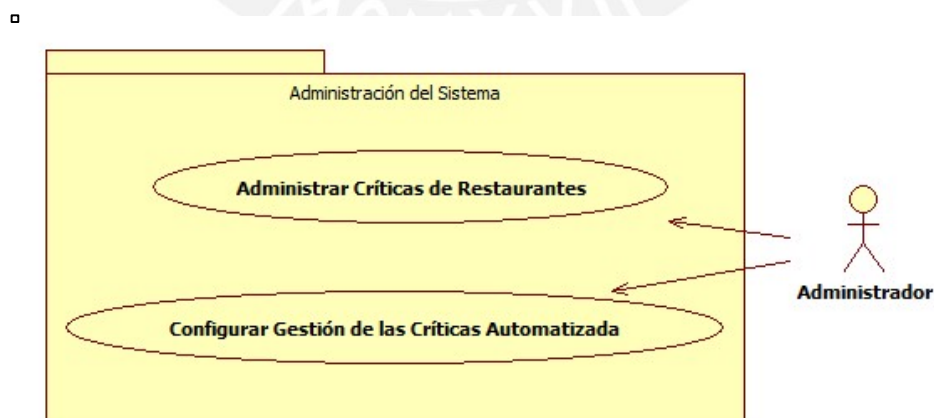


Figura 4.1 "Paquete con mecanismo de análisis de texto" (Elaboración Propia)

Por otro lado, los datos del sistema que serán utilizados por el algoritmo serán las palabras que se considerarán no apropiadas. Siguiendo con el diagrama de clases, estos datos se obtendrían del campo “Palabra” de la clase “Opcion” que, a su vez, se refleja en el diagrama de base de datos. Estas palabras serán consultadas por el servidor web para que el algoritmo de Fuerza Bruta las tome como patrones en su ejecución.

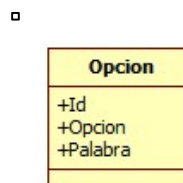


Figura 4.2 "Clases involucradas con mecanismo de texto" (Elaboración Propia)

2 Algoritmo de Búsqueda de Cadenas de Texto “Fuerza Bruta”

El algoritmo de Fuerza Bruta es el que se utilizará para la búsqueda de texto en las críticas con mensajes malintencionados. Al final de esta descripción, se puede visualizar el pseudocódigo del algoritmo adaptado para la solución que se está presentando. Para su adaptación, se tomó como base el diagrama de clases mostrado en el capítulo 2 y las referencias del algoritmo.

En el algoritmo presentado, se ordena el texto a analizar en un arreglo de segmentos. Cada segmento será una palabra a empatar con el patrón de búsqueda. El algoritmo consta de dos bucles. El primero permitirá recorrer los segmentos del texto, mientras que el segundo estará encargado del análisis de cada segmento en particular. Cada segmento pasa por cuatro verificaciones. La primera consiste en comparar la primera letra del segmento con la primera del patrón; la segunda, en comparar la última letra del segmento con la última del patrón; la tercera, en comparar la letra del medio del segmento con la correspondiente en el patrón. Finalmente, la última verificación es el bucle que analiza todas las otras letras restantes. El valor que devuelva el algoritmo será la posición del segmento que empató con el patrón, o en caso contrario, un valor negativo para indicar que no se encontró dicho patrón en el texto.

```

□ #Arreglo de segmentos s[0..n-1]
  #Patrón a buscar p

  i = 0
  while (i < n){
    if (p[0] == s[i][0]){
      if (p[long(p)-1] == s[i][long(s[i])-1]){
        if (long(p) == 2 or p[long(p)/2] == s[i][long(p)/2]){
          j = 1
          flag = true
          while ( (j < long(s[i]) - 1) and flag==true){
            if (p[j] != s[i][j])
              flag = false
            j++
          }
          if (flag == true)
            return i
          else
            return -1
        }
      }
    }
    i++
  }
  return -1

```

En la figura 4.3, se muestra la implementación del algoritmo. En esta se puede apreciar el uso de funciones como “ToUpperInvariant” para que no haya diferenciación entre mayúsculas y minúsculas en las palabras. Asimismo, se cuenta con una tabla maestra que permite llevar un registro de las palabras patrones (una de las entradas del algoritmo), es decir, un registro de las palabras que de ser incluidas en los texto de las críticas, estas serían rechazadas por mal contenido. En cuanto a los segmentos (primer parámetro del algoritmo), es un arreglo de textos, este se forma separando las palabras del comentario del comensal a analizar a través de la función ‘split’ que es una función propia de las cadenas de texto en .NET.

```

public static int FuerzaBruta(string[] segmentos, string patron)
{
    int i = 0;
    int n = segmentos.Count();
    while (i < n)
    {
        try
        {
            if (char.ToUpperInvariant(patron[0]) == char.ToUpperInvariant(segmentos[i][0]))
            {
                if (char.ToUpperInvariant(patron[patron.Length - 1]) == char.ToUpperInvariant(segmentos[i][segmentos[i].Length - 1]))
                {
                    //segmentos[i][patron.Length/2], realizar la excepción de que salga de su alcance
                    if (patron.Length == 2 ||
                        char.ToUpperInvariant(patron[patron.Length / 2]) == char.ToUpperInvariant(segmentos[i][patron.Length / 2]))
                    {
                        int j = 1;
                        bool flag = true;

                        while ( ( j < segmentos[i].Length - 1) && flag)
                        {
                            //segmentos[i][j], realizar la excepción de que salga de su alcance
                            if (char.ToUpperInvariant(patron[j]) != char.ToUpperInvariant(segmentos[i][j]))
                                flag = false;
                            j++;
                        }
                        if (flag)
                            return i;
                        else
                            return -1;
                    }
                }
            }
            i++;
        }
        catch (IndexOutOfRangeException e)
        {
            i++;
        }
    }
    return -1;
}

```

Figura 4.3 “Implementación Algoritmo Fuerza Bruta” (Elaboración Propia)

CAPÍTULO 5: DISEÑO Y CONSTRUCCIÓN

Este capítulo se centra en desarrollar y resolver el último objetivo específico, el cual resulta en la implementación del prototipo funcional. Para esto se empezará especificando algunos elementos del diseño tales como el modelo de base de datos para visualizar cómo la información se encuentra integrada físicamente y las interfaces gráficas del sistema para que pueda visualizarse el estándar que este seguirá para su construcción. Seguido de esto, se detallará el uso que se realizará sobre las herramientas para la implementación que se mencionaron en el apartado de metodologías en el capítulo 1 de generalidades. Además, se indicarán algunos patrones de programación que se utilizarán en la elaboración del sistema. Finalmente, se explicará la estrategia de pruebas elegida, los criterios de aceptación y los resultados esperados de las pruebas.

1 Diagrama de Bases de Datos

En la figura 5.1, se muestra el diagrama de base de datos, para su elaboración se tomó como principal entrada el diagrama de clases elaborado en el capítulo 2 de análisis. Asimismo, para su desarrollo se utilizó una herramienta visual de Visual Studio 2010 que permite realizar diagramas de clases. Una vez que el diagrama se encuentra elaborado, dicha herramienta brinda la opción de generar un script para SQL Server para crear una base de datos que refleje dicho diagrama. En este diagrama, se puede visualizar el uso de una tabla, sin relación con ninguna otra, denominada “Opciones”. Esta permitirá administrar las palabras que servirán de patrones a ser reconocidos en los textos de las críticas para decidir si aceptarlas o rechazarlas de forma automática por el algoritmo. Por otro lado, en las tablas de restaurantes y platos, se puede visualizar unos campos denominados “imageData” e “imageMimeType”, juntos estos dos permiten almacenar imágenes por cada registro realizado en dichas tabla. De esta forma se evita estar manejando un gestor de archivos externo, y se centraliza todo en la misma base de datos.

En cuanto a la normalización de la base de datos, esta sigue las tres primeras formas normales, con excepción en el caso de la tabla de “Estado”. Esto se debe a que se busca evitar trabajar con muchas tablas de estado por cada distinto tipo de entidad. Por esto, se definió una sola que contenga un campo que permita reconocer a qué entidad le corresponde, este es el campo “tipo”.

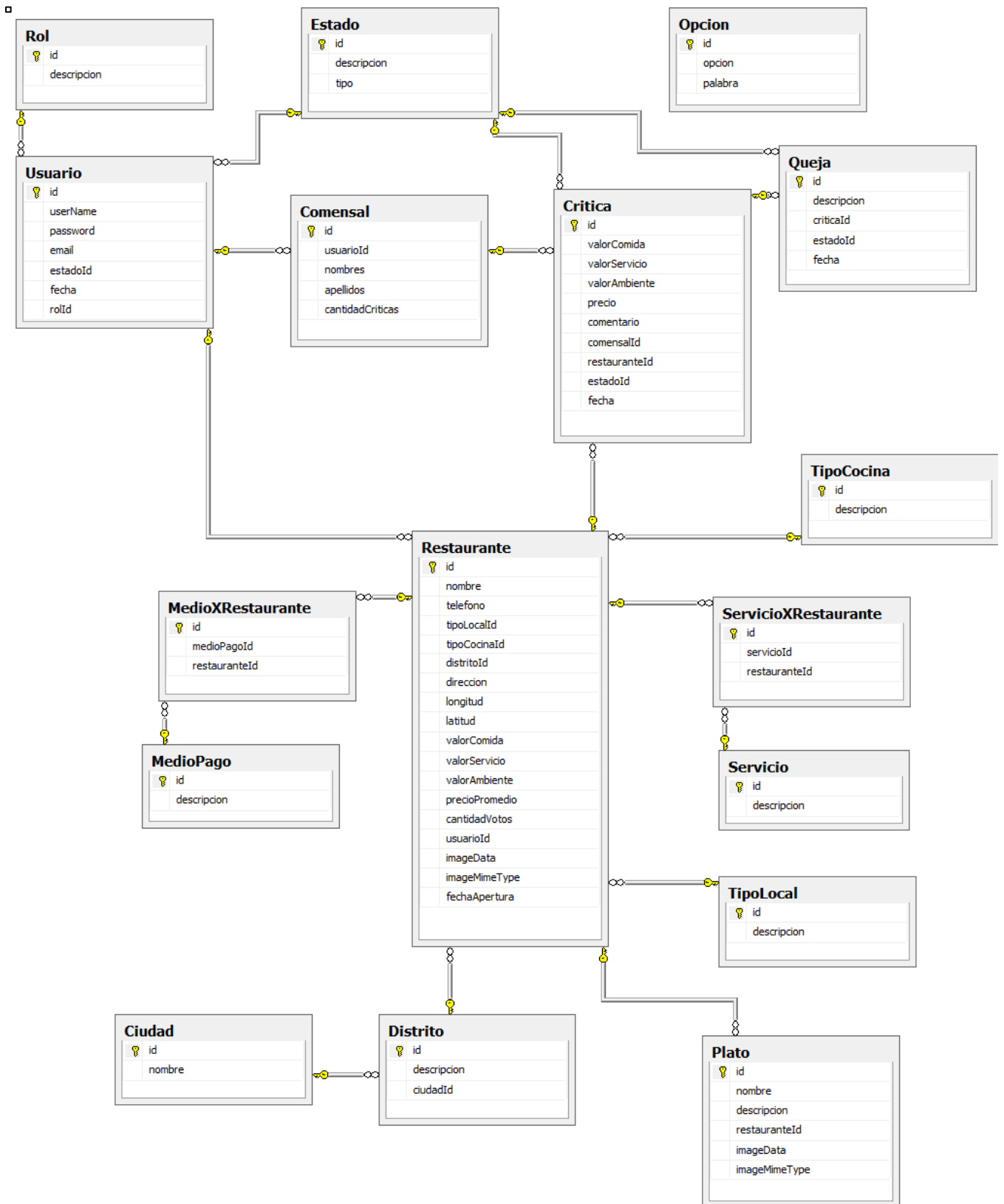


Figura 5.1 “Diagrama de Base de Datos” (Elaboración Propia)

2 Diseño de Interfaz Gráfica

En esta sección se describirán los componentes gráficos que se utilizarán en las interfaces con el cliente con el objetivo de poder estandarizar las páginas del sistema web para que al usuario se le haga más sencillo navegar por este. Asimismo, se mostrará el diseño que tendrá la aplicación tanto para los usuarios públicos, consumidores, administradores y, por último, los administradores de este.

2.1 Interfaz Pública y de Consumidores

En esta sección, se mostrará el diseño que se ha desarrollado tanto para los usuarios públicos y los consumidores.

En la figura 5.2, se puede visualizar la distribución general que tendrán las distintas páginas por las que navegará el usuario tanto público como consumidor. Como se muestra, el diseño es bastante sencillo, ya que las funciones que realiza este tipo de usuario son básicamente la de buscar restaurantes y criticarlos, las cuales son sencillas desde el punto de vista gráfico.

En la figura 5.3, se muestra una interfaz de cómo se vería la funcionalidad de consultar restaurantes que se expuso en los casos de uso del capítulo 2. Esta contiene la cabecera que siempre brindará la opción de poder iniciar sesión o desconectarse de esta. Asimismo, un menú de navegación que será siempre visible, el cuerpo principal de la página y el pie de página.

En la figura 5.4, se muestra una interfaz con un cuerpo más detallado. En este se puede visualizar algunos componentes característicos del sistema como son los mapas, las imágenes, secciones con los comentarios, entre otros.

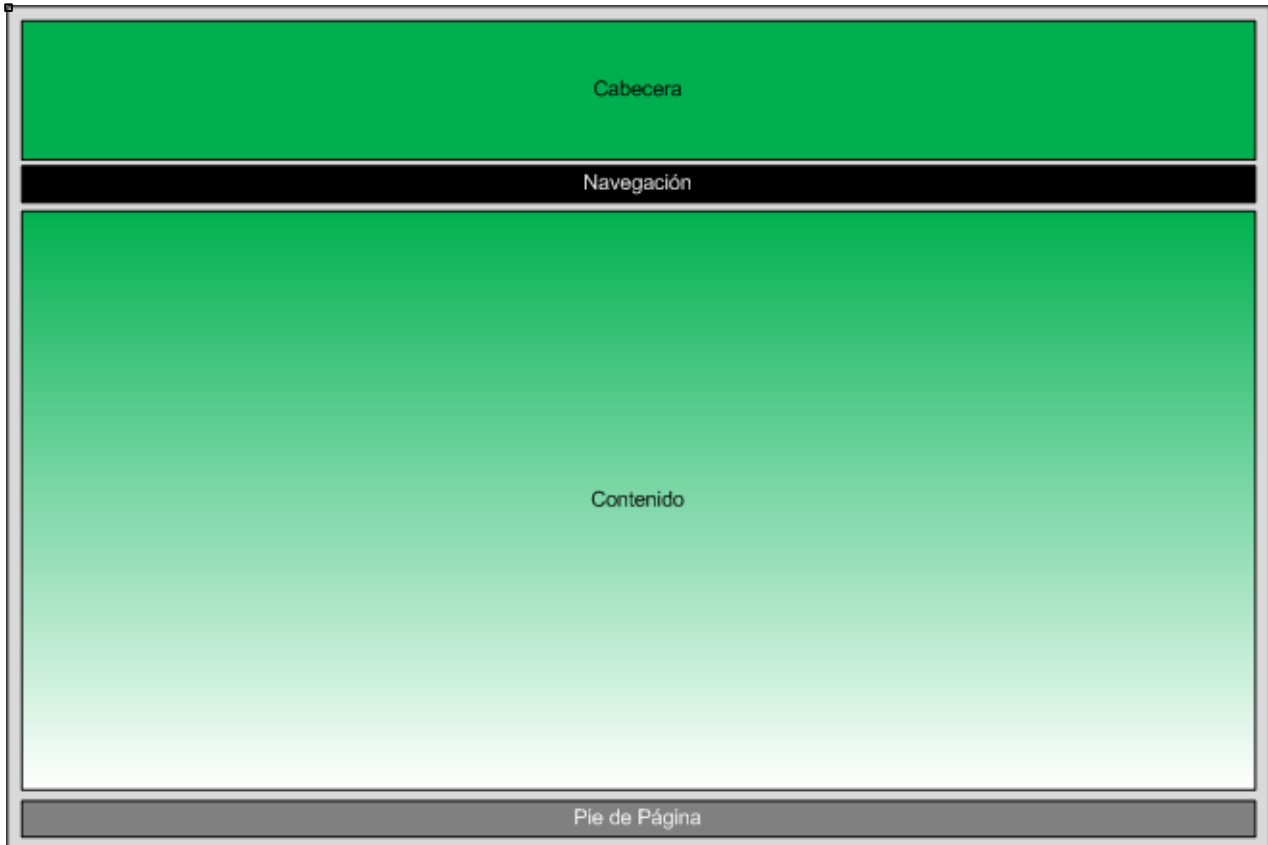


Figura 5.2 "Interfaz Pública" (Elaboración Propia)



Figura 5.3 "Consulta Restaurantes" (Elaboración Propia)

2.2 Interfaz de Administradores y Restaurantes

En esta sección, se mostrará el diseño que se ha desarrollado para los usuarios administradores y restaurantes.

En la figura 5.5, se puede visualizar la distribución general que tendrán las distintas páginas por las que navegará el usuario tanto administrador como restaurante. A diferencia de la interfaz pública, a este se le añade una sección que contiene las opciones, ya que los usuarios tendrán una mayor cantidad de funcionalidades, por lo que este “Menú de Opciones” les permitirá navegar con mayor facilidad.

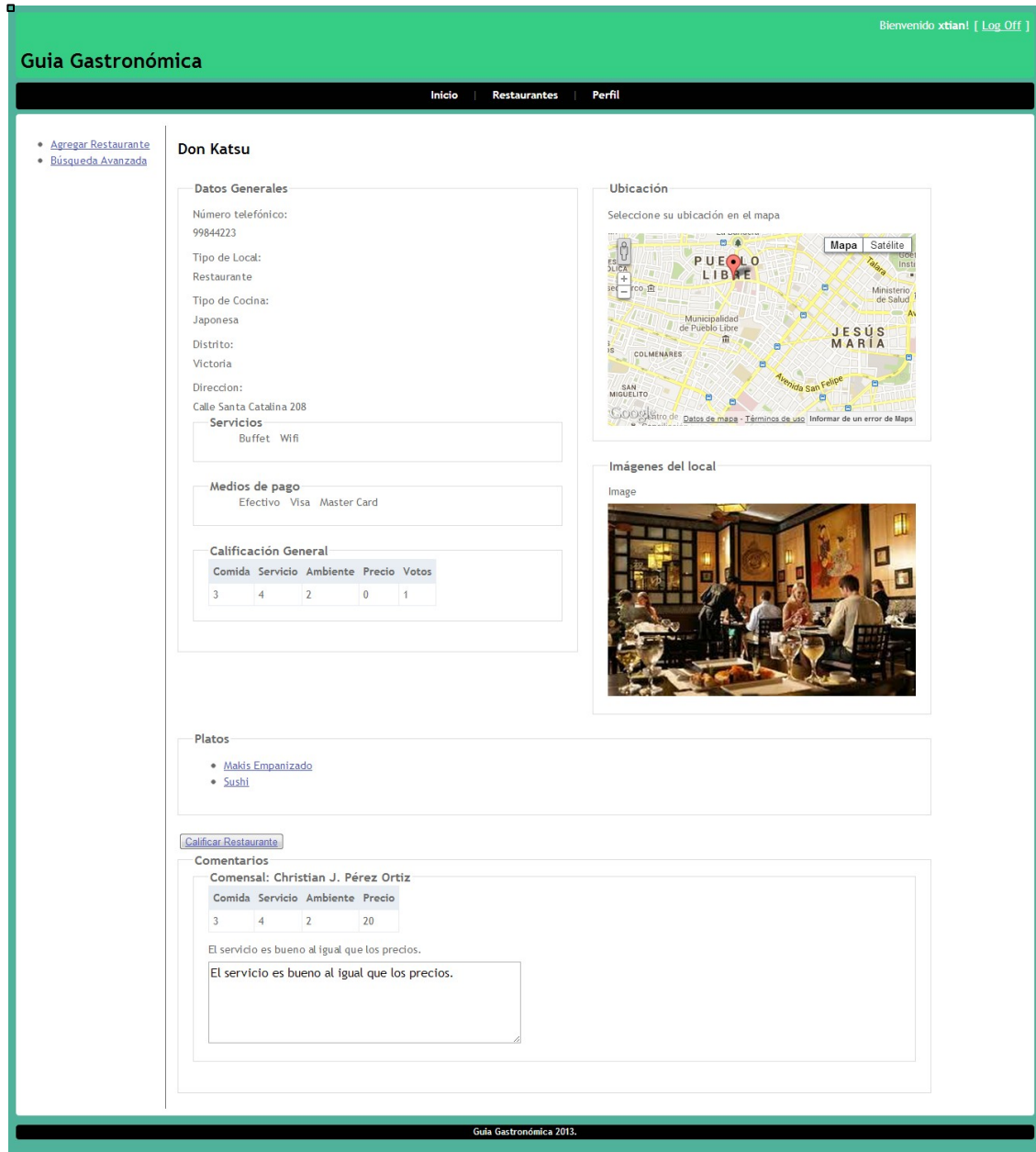


Figura 5.4 "Detalle Restaurante" (Elaboración Propia)

En la figura 5.6, se puede visualizar el diseño de la funcionalidad “Administrar Catálogo de Platos”, la cual se mencionó en los casos de uso del capítulo de Análisis.

Esta opción corresponde a un usuario dueño de un restaurante, y como se mencionó, tendrá a su disposición un contenedor con las diferentes opciones que este utilizará.

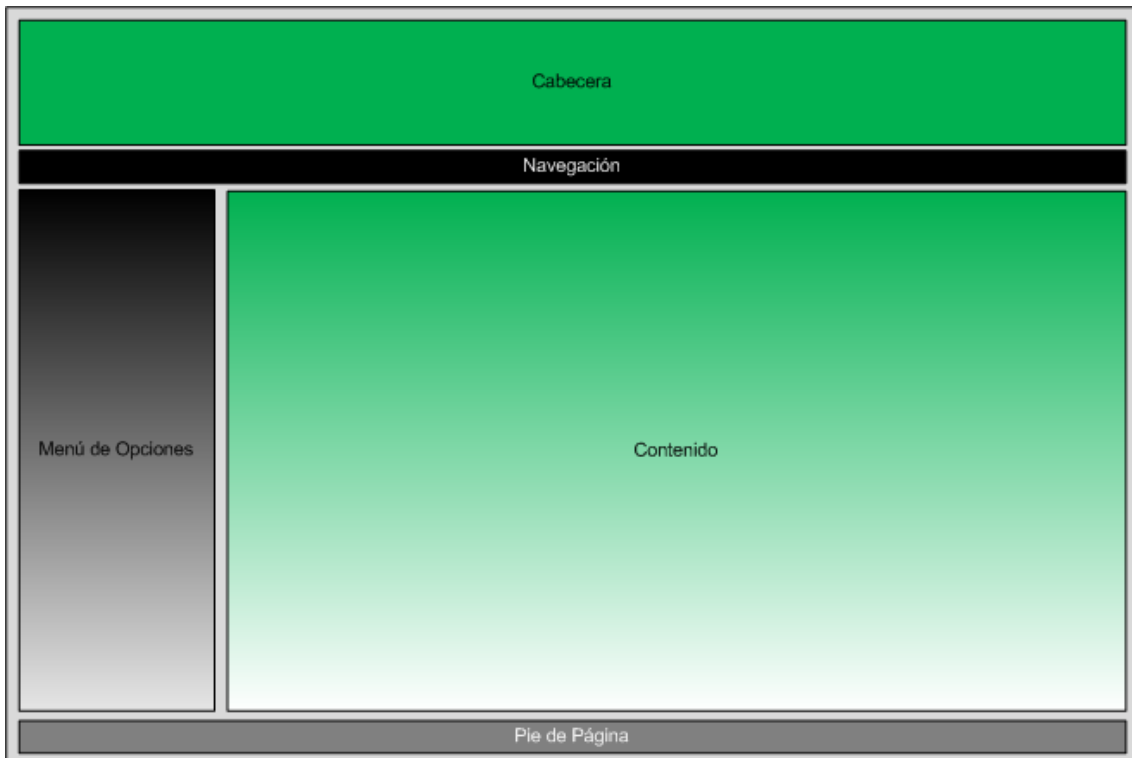


Figura 5.5 "Interfaz Restaurante - Administrador" (Elaboración Propia)

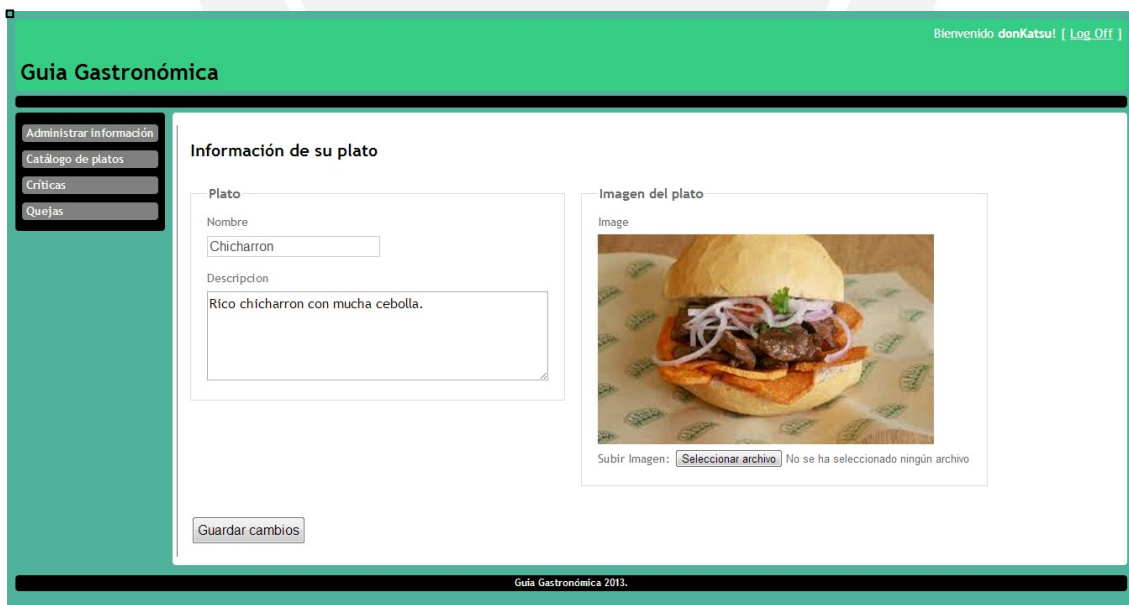


Figura 5.6 "Detalle Plato" (Elaboración Propia)

En la figura 5.7, se puede visualizar el diseño de la funcionalidad “Administrar Información del Restaurante”, el cual se mencionó en los casos de uso del capítulo de Análisis. En esta se puede observar cómo el restaurante puede modificar sus datos generales, su ubicación en el mapa y subir una foto de referencia a su gusto. Asimismo, el contenedor lateral de las opciones se sigue manteniendo visible.

Figura 5.7 "Editar Restaurante" (Elaboración Propia)

3 Construcción del Sistema

Para la construcción del sistema se seleccionó una serie de herramientas y patrones de programación. A continuación, se detalla la forma en que estos han sido aplicados a la elaboración de este sistema.

3.1 Herramientas para la implementación

Para la construcción del sistema se seleccionó una serie de herramientas y patrones de programación. A continuación, se detalla la forma en que estos han sido aplicados a la elaboración de este sistema.

Para el desarrollo del sistema se usó como plataforma principal la de .NET. Para esto, se utilizó su IDE principal, Visual Studio, en su versión 2010. Asimismo, el proyecto fue creado bajo su framework MVC3, el cual, como se mencionó en el capítulo 1, permite seguir un patrón modelo-vista-controlador. Una de las principales ventajas por las que se optó por este framework, para el desarrollo web, es que organiza el proyecto como se puede observar en la figura 5.8. En dicha figura, entre las capas que se muestra, se puede visualizar las de Modelo, Vista y Controlador. Estas permiten de una forma bastante intuitiva separar la lógica del negocio, el acceso a los datos y la forma en que se mostrará al usuario.

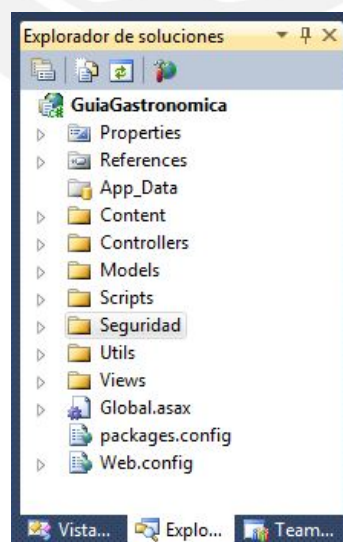


Figura 5.8 "Capas de MVC3" (Elaboración Propia)

Asimismo, el motor gráfico para generar las vistas es el denominado Razor, el cual genera un html a partir de un cshtml (el cual es una combinación de html con c# embebido en este para generar la parte dinámica del contenido). En la figura 5.9, se puede visualizar un fragmento de código del cshtml. Aquí, se puede visualizar lo anteriormente dicho, es decir, el código en c# embebido entre el html.

```

@model GuiaGastronomica.Models.Restaurante

@{
    ViewBag.Title = "Log On";
}

<h2>Log On</h2>
<p>
    Please enter your username and password. @Html.ActionLink("Register", "Register") if you don't have an account.
</p>

<script src="@Url.Content("~/Scripts/jquery.validate.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.validate.unobtrusive.min.js")" type="text/javascript"></script>

@Html.ValidationSummary(true, "Login was unsuccessful. Please correct the errors and try again.")

@using (Html.BeginForm()) {
    <div>
        <fieldset>
            <legend>Account Information</legend>
    
```

Figura 5.9 "Motor Gráfico Razor" (Elaboración Propia)

En cuanto al acceso a datos, MVC3 está bien integrado con un ORM (object oriented mapping), el denominado entity framework en su versión 5.0. Este último, permite empezar realizando el código antes de realizar la base de datos, ya que las clases creadas luego se mapearán con tablas en la base de datos. Con el uso de este ORM, el manejo de la base de datos es más sencillo porque se interactúa con esta mediante clases y ya no de una forma directa, lo cual le da una capa de abstracción al programador más sencilla de manejar.

Por otro lado, el motor de base de datos para la persistencia del sistema, es SQL Server 2008, la razón por la que se eligió este motor para el desarrollo del sistema es que al ser parte de la plataforma .NET, permite realizar el manejo de la base de datos con mayor facilidad. Adicionalmente, se utilizó la tecnología Telerik, la cual se enfoca en brindar componentes típicos de una web para no partir desde cero y poder reutilizarlos dándole así un estándar al sistema.

Por último, la plataforma .NET no es una herramienta gratuita; sin embargo, para esta tesis se está utilizando una licencia académica, la cual es gratuita por lo que no

implicará un costo adicional en el proyecto ninguna adquisición de licencias.

3.2 Patrones de programación

Los patrones de programación usados, en el sistema, han proporcionado un estándar para implementar ciertas capas o funciones del sistema.

Para este proyecto en particular, dos patrones se han utilizado básicamente: el patrón Modelo-Vista-Controlador (MVC) y el patrón Data Access Object (DAO).

- **Patrón MVC**

El patrón MVC permite dividir la aplicación en capas lógicas independientes entre ellas. Esto permite que la implementación se encuentre mejor organizada y apoye al desarrollador en su tarea. Las capas son organizadas en tres, las cuales se detallan a continuación.

Por un lado, se encuentra la capa de Modelo. Esta contiene la información de la aplicación. En esta capa se incorporan los datos, las reglas de validación y el acceso a los datos. Como se mencionó en la sección anterior, MVC3 de .NET, ya brinda un entorno predispuesto para este patrón, por lo que para el proyecto, en esta capa se colocarían todas las entidades del sistema con un acceso a los datos por medio de entity framework.

Por otro lado, se encuentra la capa de Vista. Esta encapsula la presentación, es decir, la interfaz final que será enviada al usuario. Para el caso de MVC3, las vistas se empaquetan por controlador definido, y cada una se genera a partir de los métodos que se declaran dentro de dichos controladores.

Finalmente, se encuentra la capa Controlador. Esta contiene la lógica que controla el flujo del programa. Para esto, este se encuentra pendiente de los cambios en las vistas, y una vez detectados, se los envía al modelo, el cual le regresa una respuesta a la vista.

- **Patrón DAO**

El patrón DAO permite suministrar una interfaz entre la aplicación y un almacenamiento de datos tales como una base de datos. Una de las ventajas que se logra con este patrón es que el acceso a datos se realiza a través de objetos del negocio y no se requiere conocimiento directo del destino final de la información que manipula, por lo que se evita el uso de APIs de los distintos motores de base de datos.

En la figura 5.10, se muestra un diagrama del comportamiento de este patrón. A continuación se explica cada componente.

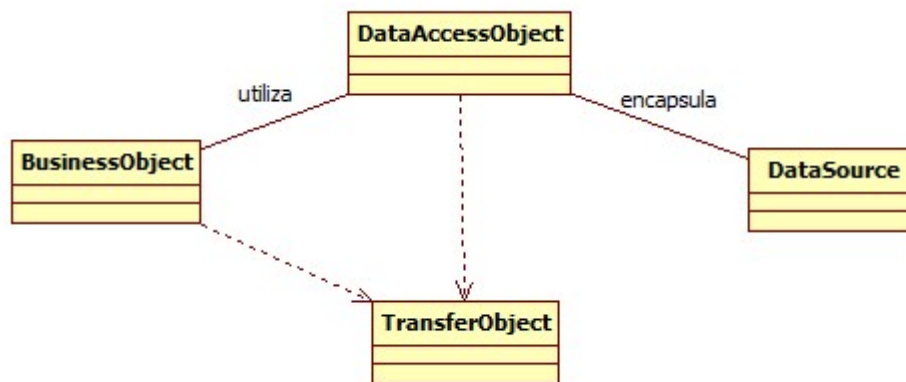


Figura 5.10 "Patrón DAO" (Elaboración Propia)

Por un lado, el **BusinessObject** u objeto del negocio es el que representa los datos y requiere acceso a la fuente de estos. En el caso de este proyecto, los objetos del negocio son las clases en C# en la capa de Modelo.

Por otro lado, se encuentra el **DataAccessObject** u objeto de acceso, el cual es la parte principal del patrón, ya que este abstrae la implementación del acceso a los datos para que el **BusinessObject** pueda acceder a ellos ignorando de qué tipo de fuente se trate.

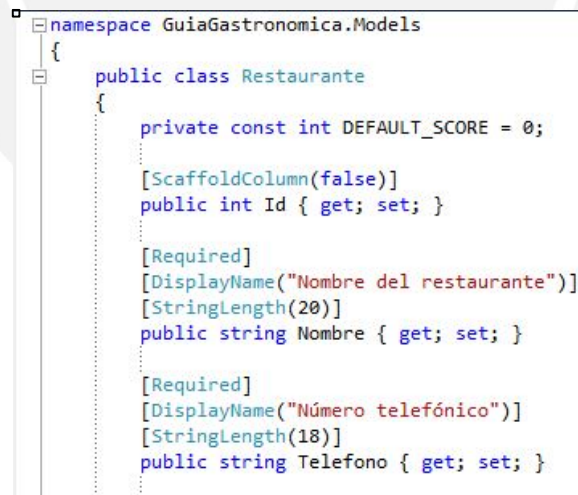
El siguiente componente es el **DataSource** o fuente de datos. Este es cualquier fuente de datos que se pueda utilizar como un motor de base de datos (que en este caso es SQL Server 2008), un repositorio, etc.

Por último, el TransferObject u objeto de transferencia es un objeto de soporte que puede ser utilizado para devolver los datos al cliente.

3.3 Estándares de Programación

Los estándares de programación utilizados tienen como fin brindar un código fuente legible y sencillo de leer. Para la elaboración de estos estándares, se ha tomado como base las indicaciones generales de las guías del framework de .NET y se ha adaptado para este proyecto. A continuación se explicarán los estándares elaborados para lo nombres de las variables, el detalle de los demás componentes se podrá observar en los anexos.

Para los nombres de funciones, propiedades y clases, se utiliza la nomenclatura de PascalCasing, la cual consta en colocar todas las palabras juntas con la primera en mayúscula, incluida la primera palabra. Un ejemplo de esto se muestra en la figura 5.11. En esta, se puede observar que las propiedades Id, Nombre y Teléfono, cumplen con la nomenclatura descrita.



```

namespace GuiaGastronomica.Models
{
    public class Restaurante
    {
        private const int DEFAULT_SCORE = 0;

        [ScaffoldColumn(false)]
        public int Id { get; set; }

        [Required]
        [DisplayName("Nombre del restaurante")]
        [StringLength(20)]
        public string Nombre { get; set; }

        [Required]
        [DisplayName("Número telefónico")]
        [StringLength(18)]
        public string Telefono { get; set; }
    }
}

```

Figura 5.11 "Estándar Propiedades" (Elaboración Propia)

Por otro lado, en cuanto a las variables miembro, los parámetros o las variables locales, se utilizará la nomenclatura camelCasing, la cual consta en colocar todas las palabras juntas con la primera letra en mayúscula, pero a diferencia del PascalCasing, la primera debe ir en minúscula. En la figura 5.12, se puede visualizar un ejemplo con un

parámetro. En esta figura, el parámetro es restaurante, como se puede observar, su primera letra es en minúscula, si esta consistiera de dos palabras, la siguiente tendría la primera en mayúscula.

```

[HttpPost]
public ActionResult Edit_Admin(Restaurante restaurante)
{
    if (ModelState.IsValid)
    {
        db.Entry(restaurante).State = EntityState.Modified;
        db.SaveChanges();
        var cuenta = db.Accounts.Single(a => a.Id == restaurante.AccountModelID);
        cuenta.EstadoID = restaurante.EstadoID;
        db.Entry(cuenta).State = EntityState.Modified;
        db.SaveChanges();

        return RedirectToAction("Index_Admin");
    }
    ViewBag.AccountModelID = new SelectList(db.Accounts, "Id", "UserName", restaurante.AccountModelID);
    return View(restaurante);
}

```

Figura 5.12 "Estándar Parámetros" (Elaboración Propia)

3.4 Reutilización de Código

En cuanto a la reutilización de código se aprovechó algunos mecanismos propios del framework de .NET y otros que conciernen a las hojas de estilo y los scripts.

Por un lado, se reutilizó el mecanismo de seguridad de Authentication Forms que brinda el framework. Este consta de todas las funcionalidades básicas de seguridad (autenticación, registro y cambio de contraseña). De tal forma que el esfuerzo consistió en integrar este mecanismo con la tabla de usuario de la guía gastronómica.

Por otro lado, se reutilizó algunos mecanismos de las vistas que provee el motor gráfico de Razor, los cuales son las "Vistas Parciales" y las "Plantillas de Vistas". A través de las vistas parciales, se generaron los menús para los usuarios administradores, restaurantes y comensales, de tal forma que solo fuera necesario definirlos una vez para que sean navegables permanentemente a través de las distintas funcionalidades. En cuanto a las plantillas de vistas, se utilizaron para crear pequeños componentes genéricos que podrían repetirse a lo largo de las vistas. Un ejemplo es el menú de checkbox que se tiene por los servicios y medios de pago que se pueden visualizar en la figura 5.7. Estos fueron creados por una plantilla y la misma se reutiliza en otras vistas.

Finalmente, se hizo reutilización de estilos y scripts. En cuanto a los estilos se hizo provecho de su característica de poder asignar clases de estilos a diferentes etiquetas del html para así implementar el estilo una vez y repetirlo a lo largo del sistema. Por último, en cuanto a los scripts, se reutilizaron los scripts que hacían uso de los mapas de google, ya que las funcionalidades que se hacían de ellos eran casi las mismas en las distintas vistas.

4 Pruebas

En esta sección del capítulo, se detallarán qué tipos de pruebas serán ejecutadas para poder aceptar el prototipo funcional. Asimismo, se definirá una estrategia para poder decidir cuando una funcionalidad del sistema se encuentra en un umbral de aceptación o de rechazo, por lo que a continuación se presentarán los tipos de casos de prueba a utilizar y los criterios de aceptación.

4.1 Planteamiento de casos de prueba

En este punto, se mostrarán los tipos de pruebas que se utilizarán en esta parte final de la elaboración del prototipo funcional. Asimismo, esta sección permitirá divisar cuáles son las faltas en el desarrollo en caso hubiese alguna, y así se conocerá si se está cumpliendo con lo planteado durante el capítulo 1 y 2.

- **Pruebas Unitarias**

Las pruebas unitarias están enfocadas a evaluar unidades del sistema. En el caso de este proyecto, en el capítulo 2, este fue distribuido en varios paquetes, por lo que para las pruebas unitarias se verificará que cada uno de estos paquetes por separado esté cumpliendo con lo indicado en los requerimientos del sistema.

- **Pruebas de Caso de Uso**

Las pruebas de caso de uso se acoplan con las pruebas unitarias, ya que estas se

dividirán en pruebas de caso de uso, las cuales son las que finalmente serán probadas y mapeadas para corroborar que se está cumpliendo con las especificaciones del sistema.

- **Pruebas de Integración**

Una vez finalizadas las pruebas unitarias, se pasará a realizar las pruebas de integración que se constituye en realizar pruebas que aborden funcionalidades a lo largo de los distintos paquetes del sistema, es decir, el flujo completo.

4.2 Identificación de los casos de prueba

Para la identificación de los casos de prueba se parte de los puntos más particulares del sistema que son los casos de uso. De esta forma, los casos de prueba se mapearán con cada uno de estos. Una vez identificados todos, se pasará a agruparlos por paquetes para poder identificar los casos de pruebas unitarios. Finalmente, los casos de pruebas de integración se identificarán a través de las funcionalidades que aborden todos los paquetes del sistema. Para la ejecución de cada caso de prueba, se tendrá en consideración el actor y la acción que este realizará frente a qué resultado esperado. Si durante la ejecución de pruebas se obtienen los resultados esperados, se registrarán los resultados exitosos, en caso contrario, se registrará la especificación de cómo falló el caso de prueba.

4.3 Entorno de Prueba

En cuanto al entorno de prueba, se realizará en el explorador de Chrome en su versión 23, ya que se indicó como un requerimiento no funcional en el capítulo 2. Al ser un producto web, el sistema operativo bajo el cual se pruebe no influirá en los casos de prueba.

Por otro lado, en cuanto al hardware, las pruebas se realizarán desde el lado del servidor y del cliente en un entorno con las siguientes características:

Servidor	Cliente
AMD Phenom II X3 N830	Procesador Core 2 Duo
6 GB DDR3 RAM	2 GB DDR2 RAM

4.4 Criterios de aceptación o rechazo

Los criterios de aceptación o rechazo se basan en la lista de requerimientos que corresponde al capítulo 2. Por lo tanto, en la medida que los requerimientos se estén cumpliendo al formar parte de un caso de prueba y este devolver un resultado positivo, las pruebas serán aceptadas y se seguirá de esa forma hasta ejecutar todo el catálogo de pruebas.

4.5 Plantilla de las Pruebas

En la siguiente tabla, se muestra la plantilla de las pruebas a ejecutar. Esta muestra el caso de prueba que se está tratando, así como el caso de uso con el que se relaciona. Asimismo, se muestran los flujos que se probarán, las acciones y los resultados esperados por cada uno de estos.

Nombre de la Prueba Formal		
Datos	Id	TC-CON-001
	Descripción	Esta prueba consiste en ...
	Referencia	CU-CON-001
	Caso de Uso	
	Paquete	Flujo Consumidores
	Actores	Consumidor, Usuario Público.
Prueba Flujo 1: “Nombre del Flujo”		
Acción del actor		Resultado Esperado
1. El actor ...		2. El sistema muestra ...
Prueba Flujo 2: “Nombre del Flujo”		
1. El actor ...		2. El sistema muestra ...

Por otro lado, en la siguiente tabla, se muestra la plantilla que se utilizará para registrar los resultados de la ejecución formal de las pruebas. En esta plantilla, se registra la referencia al caso de prueba ejecutado, así como los resultados y en qué fase de las pruebas se encuentra.

Nombre del Resultado de la Prueba Formal		
Datos	Id	TCR-CON-001
	Referencia	TC-CON-001
	Caso de Prueba	
	Fase de Prueba	Unitaria/Integral
	Resultado	Descripción...
	Fecha	mm/aaaa

4.6 Resultados de las Pruebas

La ejecución de las pruebas permitió hallar unos errores menores en el software, los cuales fueron reparados. A continuación se da un resumen de dicha ejecución.

En cuanto a las pruebas de casos de uso, se aseguró durante el desarrollo que las especificaciones se estén cumpliendo, por lo que la ejecución de estas pruebas resultó en una aceptación completa.

Por otro lado, en cuanto a la ejecución de las pruebas unitarias (o de paquetes como se está considerando), se hallaron algunas inconsistencias en el paquete de consumidores: los “usuarios públicos” tenían acceso a funcionalidades que no formaban parte de su rol sino a la del usuario consumidor.

Finalmente, en cuanto a las pruebas integradas, se aseguró el correcto funcionamiento de distintas combinaciones en los usos que podía darse entre paquetes. Debido a que los paquetes son usados por perfiles distintos, dio un resultado similar a las pruebas unitarias, por lo que no se encontró errores.

CAPÍTULO 6: CONCLUSIONES Y RECOMENDACIONES

En este último capítulo, se presentan las conclusiones obtenidas frente a los resultados esperados que se alcanzaron a lo largo del desarrollo del proyecto. Asimismo, se expone ciertas recomendaciones para trabajos que puedan darse en el futuro y que tomen como base el estudio presentado aquí y sus conclusiones.

1 Conclusiones

En base a los resultados esperados obtenidos a lo largo del desarrollo de este proyecto, se ha concluido lo siguiente:

- En el capítulo 2, se ha especificado una arquitectura para la plataforma web que le permitirá a esta ser sostenible y escalable en el tiempo, ya que a través de las distintas vistas de arquitectura, se entiende cómo debe funcionar el sistema lógicamente y, además, se conoce los componentes que lo conforman en la implementación. Así, si en un futuro se desea continuar con la plataforma, agregando nuevas funcionalidades que luego deberán ser implementadas, se tiene esta arquitectura como una guía esta continuidad.
- En el capítulo 3, se ha construido un mecanismo de ordenamiento para las distancias entre los restaurantes y los comensales, de tal forma que la plataforma pueda aportar un nuevo criterio de evaluación con el cual los comensales podrán decidir qué restaurante se acomoda mejor a sus necesidades, por lo que da valor agregado a la guía gastronómica. Asimismo, en este capítulo se analizaron dos formas válidas de obtener las distancias mencionadas que, para el alcance de este proyecto, ambas fueron útiles, pero para una implantación real del sistema lo recomendable sería utilizar los servicios de distancia de Google. Finalmente, si se quisiera cambiar el algoritmo de ordenamiento utilizado, por distintas razones, la sección de integración con el sistema ha dejado claro cómo realizar dichos cambios.
- En el capítulo 4, se ha construido un artefacto de análisis de texto que será puesto en uso automáticamente cada vez que se registre una crítica por parte de los

comensales, para este se tomó como base la adaptación de un algoritmo de búsqueda de cadenas de texto que permite reconocer si los textos de las críticas contienen palabras ofensivas. Este artefacto aporta una nueva forma de analizar las críticas, agilizando su proceso de publicación, puesto que no será necesaria la revisión del administrador del sistema. De esta forma, este artefacto da valor agregado a la guía gastronómica. Finalmente, si se quisiera cambiar el algoritmo de análisis de texto utilizado, por distintas razones, la sección de integración con el sistema ha dejado claro cómo realizar dichos cambios.

- En el capítulo 5, se consiguió implementar un prototipo funcional del sistema de información. El principal aporte de este ha sido poder demostrar ser una alternativa de solución a la problemática planteada, la cual se centra en la falta de mecanismos de ubicación de restaurantes y en la falta de un artefacto que apoye en el análisis de los comentarios. Así, se logró crear un espacio en el que se mantenía un registro de restaurantes y comensales, en el cual estos últimos son capaces de encontrar los locales que se adecúen mejor a sus necesidades y criticarlos para generar información neutral con respecto a los servicios que brindan. Asimismo, este prototipo permitió integrar los mecanismos de ordenamiento y de análisis de texto que se mencionaron previamente.

2 Recomendaciones

Este proyecto de fin de carrera podría tomarse como base para futuros trabajos. A continuación se menciona las recomendaciones de cómo podría extenderse el estudio de este proyecto:

- Incluir funcionalidades de ruteo en los mapas de Google integrados con la guía gastronómica para dar una mejor experiencia al usuario de cómo llegar hacia el restaurante que le interesa.
- Incluir una funcionalidad que permita registrar pedidos de delivery para que solo por medio de la consulta de un restaurante en la guía gastronómica se puede solicitar el

pedido, de tal forma que se brinde al comensal una opción diferente a la tradicional de realizar los pedidos vía llamada.

- Incluir interfaces para realizar transacciones de pago en caso se quiera realizar un pago por delivery mediante la web, de tal forma que los pagos se registren en los sistemas de los restaurantes, y así facilitar este procedimiento entre restaurantes y comensales.
- Realizar investigaciones acerca de los algoritmos de ordenamiento y de búsqueda de cadenas de textos para encontrar otros que sean más eficientes y se adapten mejor al sistema, para que así se mejore la experiencia de los usuarios.
- Incluir funcionalidades que permitan consumir los servicios web de empresas de cupones para así poder consultar aquellos cupones que sean de restaurantes y poder publicarlos en la guía gastronómica de forma que los comensales puedan visualizarlos, y así tener conocimiento de las ofertas vigentes.
- Realizar un estudio de SEO (search engine optimization) para que, una vez publicada la plataforma web, se empiece a generar tráfico de usuarios.

REFERENCIAS BIBLIOGRÁFICAS

ALI, Rawan

2011 “An Algorithm for String Searching Based on Brute-Force Algorithm”.
International Journal of Computer Science and Network Security. EEUU:
volumen 11, número 7.

AGUILAR, Eddie

2011 SEO: Posicionamiento Y Optimización de Sitios Web. Edición Española.
México D.F.: Editorial Limusa S.A. de C.V.

BOYER, Robert y MOORE, Strother

1977 “A Fast String Searching Algorithm”. Association for Computing
Machinery, Inc. EEUU, volumen 20, número 10.

BRUCE, Silver

2011 “BPMN method and style: with BPMN implementer’s guide”. Segunda
edición. California: Cody-Cassidy Press.

CUEVAS, Francisco

2004 “Control de costos y gastos en los restaurantes”. Edición Española.
México D.F.: Editorial Limusa S.A. de C.V.

DEGUSTA

2012 Degusta. Consulta: 20 de setiembre del 2012.
<<http://www.degusta.com.pe/>>

EMPRESA EDITORA EL COMERCIO S.A.

2009 “Sólo 800 de 220 mil restaurantes de Lima tienen certificación de
saludables”. La Gestión. Lima, 17 de mayo. Consulta: 20 de setiembre
del 2012.

<<http://gestion.pe/noticia/287810/solo-800-220-mil-restaurantes-lima-tienen-certificacion-saludables>>

EVANS, Dave

2008 Social Media Marketing “An Hour a Day”. Primera Edición.
Indianapolis: Wiley Publishinc, Inc.

FOURSQUARE

2012 Foursquare. Consulta: 20 de setiembre del 2012.
<<https://foursquare.com/>>

GOODWIN, H.

1910 “The Haversine in Nautical Astronomy”. Naval Institute Proceedings.
Volumen 36, número 3, pp. 735–746.

GPS RESTAURANTES

2012 GPS Restaurantes. Consulta 25 de octubre del 2012.
<http://guiagps.clasificados.pe/restaurantes/guia_restaurantes_gourmet>

GUÍA OLEO S.A.

2012 Guía Oleo. Consulta 25 de octubre del 2012.
<<http://www.guiaoleo.com.ar/acercadeguiaoleo>>

INEI

2013 “Informe Técnico: Las Tecnologías de Información y Comunicación en
2013 los Hogares”. N° 01. Lima, marzo. Consulta: 07 de abril del 2013.
<<http://www.inei.gov.pe/web/Biblioinei/BoletinFlotante.asp?file=16146.pdf>>

INEI

2012.a “Nota de Prensa”. n°140. Lima, agosto. Consulta: 20 de setiembre del
2012. <<http://www.inei.gov.pe/web/NotaPrensa/Attach/14813.pdf>>

INEI

2012.b “Ventas Reales del Sector Restaurantes: 2011-2012”. Lima, junio.
Consulta: 22 de setiembre de 2012.
<http://proyectos.inei.gov.pe/emcrs/Pag_Resultados2009.asp?varcuadro=1b>

- INEI
2011 “Nota de Prensa”. n°030. Lima, febrero. Consulta: 20 de setiembre del 2012. <<http://www.inei.gob.pe/web/NotaPrensa/Attach/12079.pdf>>
- INEI
2010 “Perú: Informe Económico Mensual de 2010 Febrero”. Única Edición. Centro de Edición de la OTD.
- INEI
2008 “Censos Nacionales 2007: XI de Población y VI de Vivienda”. Lima, setiembre. Consulta: 07 de abril del 2013. <<http://desa.inei.gob.pe/censos2007/tabulados/>>
- IPM
2011 “Marketing para restaurantes”. Consulta: 02 de noviembre del 2012. <http://www.ipm.com.pe/mktg_res.htm#ini>
- KNUTH, Donald y MORRIS, James y PRATT, Vaughan
1977 “Fast Pattern Matching in Strings”. SIAM Journal on Computing. Volumen 2, número 2.
- KRUTCHEN, Philippe
1995 “Architectural Blueprints – The 4+1 View Model of Software Architecture”. IEEE Software. Volumen 1, número 6, pp. 42-50.
- LAUDON, Kenneth y Carol GUERCIO
2009 e-commerce: negocios, tecnología, sociedad. Cuarta Edición. México D.F.: Pearson Prentice Hall.
- LAUDON, Kenneth y Jane LAUDON
2008 Sistemas de Información en negocios. Décima Edición México D.F.: Pearson Prentice Hall.
- MIRANDA, Loretta y Liseth SÁNCHEZ
2010 Estudio de Pre-Factibilidad para la Implementación de un Restaurante de Comida en Base a Carne de Cuy. Tesis para optar el Título de Ingeniero

Industrial. Lima: Pontificia de Universidad Católica del Perú, Facultad de Ciencias e Ingeniería.

ORONSKY, Carl y Prakash CHATHOTH

2007 “An exploratory study examining information technology adoption and implementation in full-service restaurant firms”. *International Journal of Hospitality Management*. 2007, volume 26, topico 4, 941-956. Consulta: 22 de setiembre del 2012.

<<http://www.sciencedirect.com.ezproxybib.pucp.edu.pe:2048/science/article/pii/S0278431906000326>>

PEARL, Judea

1984 “Heuristics: intelligent search strategies for computer problem solving”. Primera Edición. EEUU: Addison-Wesley Longman Publishing Co., Inc. Boston.

PER, Philippe

2003 *Rational Unified Process Made Easy: A Practitioner’s Guide to the RUP*. EEUU: Pearson Education, Inc.

PMI

2008 *Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK)*. Cuarta Edición. EEUU: Project Management Institute, Inc.

QUEREMOSCOMER

2012 QUEREMOSCOMER.COM. Consulta: 25 de octubre del 2012.

<<http://www.queremoscomer.com/quienes-somos/>>

RODRIGUEZ, Gustavo y VENTURO, Sandro

2007 “Perú es el país con más platos típicos en el mundo”. *Ampay Perú*, 357 listas para entender cómo somos los peruanos. Primera Edición. Lima: Aguilar E.I.R.L.

SHELL, Donald

1959 “A High-Speed Sorting Procedure”. *Communications of the ACM*.

Volumen 2, número 7, pp. 30-32.
<<http://dl.acm.org/citation.cfm?doid=368370.368387>>

TELEFÓNICA

2010 “Comunicaciones móviles y sociedad”. Consulta: 30 de octubre del 2012.
<sociedadinformacion.fundacion.telefonica.com/sección=1268&idioma=es_ES &id=2010072708320001&activo=6.do#>

TRIPADVISOR INC.

2012 TripAdvisor. Consulta: 1 de noviembre del 2012.
<http://www.tripadvisor.com.ar/pages/about_us.html>

REENSKAUG, Trygve

1979 "MVC: XEROX PARC 1978-79". Consulta: 09 de abril del 2013.
<<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>>

UNIVERSITY OF FLORIDA

2012 “El Sistema de Posicionamiento Global - GPS”. Consulta: 30 de octubre del 2012.
<edis.ifas.ufl.edu/in657>

WHITE, Stephen

2009 “BPMN: Guía de Referencia y Modelado”. Edición Español. EEUU: Future Strategies Inc., Book Division.

WILLIS, H.

2012 “Monkey Dish to the death!”. Restaurant Business. Volumen 111, número 9, pp. 12-14.

W3C

2015 “Geolocation API Specification”. Consulta: 7 de abril del 2015.
<<http://www.w3.org/TR/geolocation-API/>>