

## Anexos

## Contenido

1. Catálogo de actores .....	1
2. Casos de uso .....	2
3. Especificación de casos de uso.....	6
3.1 Paquete de Seguridad .....	6
3.2 Paquete de Administración .....	10
3.3 Paquete de Restaurantes .....	19
3.4 Paquete de Consumidores .....	23
4. Matriz Requerimientos – Caso de Uso.....	29
5. Diccionario de Datos .....	35
6. Interfaces .....	42
6.1 Paquete de Seguridad .....	43
6.2 Paquete de Comensales .....	44
6.3 Paquete de Restaurante.....	48
6.4 Paquete de Administración .....	51
7. Estándares .....	56
7.1 Llaves de bloque .....	56
7.2 Comentarios .....	57
7.3 Espaciado .....	57
7.4 Nombramiento.....	58
8. Estructura del proyecto.....	60
8.1 Content .....	60
8.2 Controllers.....	61
8.3 Models.....	62
8.4 Scripts.....	63
8.5 Seguridad.....	64
8.6 Utils .....	65
8.7 Views.....	66
9. Implementación de los algoritmos .....	67
9.1 Algoritmo Shell.....	67
9.2 Algoritmo Fuerza Bruta .....	69

10. Casos de Prueba .....	69
10.1 Paquete de Seguridad .....	70
10.2 Paquete de Administración .....	73
10.3 Paquete de Restaurante.....	75
10.4 Paquete de Consumidores .....	76
11. Ejecución de Pruebas .....	77
11.1 Fase Casos de Uso.....	78
11.2 Fase Unitaria .....	79
11.3 Fase Integración .....	80



## Índice de Figuras

Figura 1 “Seguridad – Casos de Uso” (Elaboración Propia) .....	2
Figura 2 “Restaurantes – Casos de Uso” (Elaboración Propia) .....	3
Figura 3 “Administración – Casos de Uso” (Elaboración Propia).....	4
Figura 4 “Consumidores – Casos de Uso” (Elaboración Propia).....	4
Figura 5 “Interfaz – Iniciar Sesión” (Elaboración Propia).....	43
Figura 6 “Interfaz – Registrar Comensal” (Elaboración Propia) .....	44
Figura 7 “Interfaz – Consultar Restaurantes” (Elaboración Propia).....	44
Figura 8 “Interfaz – Detalle de Restaurante” (Elaboración Propia).....	45
Figura 9 “Interfaz – Detalle Plato” (Elaboración Propia) .....	46
Figura 10 “Interfaz – Calificar Restaurante” (Elaboración Propia).....	46
Figura 11 “Interfaz – Consultar Perfil” (Elaboración Propia).....	47
Figura 12 “Interfaz – Administrar Información del Restaurante” (Elaboración Propia) .....	48
Figura 13 “Interfaz – Consultar Catálogo de Platos” (Elaboración Propia) .....	49
Figura 14 “Interfaz – Editar Plato” (Elaboración Propia).....	49
Figura 15 “Interfaz – Restaurante Consulta Críticas” (Elaboración Propia) .....	50
Figura 16 “Interfaz – Reportar Críticas” (Elaboración Propia).....	51
Figura 17 “Interfaz – Administrar Servicios” (Elaboración Propia).....	51
Figura 18 “Interfaz – Administrador Consulta Críticas” (Elaboración Propia).....	52
Figura 19 “Interfaz – Administrador Edita Críticas” (Elaboración Propia).....	52
Figura 20 “Interfaz – Administrador Consulta Quejas” (Elaboración Propia) .....	53
Figura 21 “Interfaz – Administrador Edita Quejas” (Elaboración Propia) .....	53
Figura 22 “Interfaz – Administrador Consulta Comensales” (Elaboración Propia) .....	54
Figura 23 “Interfaz – Administrador Edita Comensal” (Elaboración Propia) .....	54
Figura 24 “Interfaz – Administrador Consulta Restaurantes” (Elaboración Propia).....	55
Figura 25 “Interfaz – Administrador Edita Restaurantes” (Elaboración Propia).....	55
Figura 26 “Interfaz – Administración del Algoritmo” (Elaboración Propia).....	56
Figura 27 “Estándar - Llaves” (Elaboración Propia) .....	56
Figura 28 “Estándar - Comentarios” (Elaboración Propia).....	57
Figura 31 “Estructura del Proyecto - Content” (Elaboración Propia).....	61
Figura 32 “Estructura del Proyecto - Controllers” (Elaboración Propia).....	62
Figura 33 “Estructura del Proyecto - Models” (Elaboración Propia).....	63
Figura 34 “Estructura del Proyecto - Scripts” (Elaboración Propia) .....	64
Figura 35 “Estructura del Proyecto - Seguridad” (Elaboración Propia) .....	65
Figura 36 “Estructura del Proyecto - Utils” (Elaboración Propia).....	66
Figura 37 “Estructura del Proyecto - Views” (Elaboración Propia) .....	67
Figura 38 “Algoritmo - Shell” (Elaboración Propia) .....	68
Figura 39 “Fórmula Distancia” (Elaboración Propia).....	68
Figura 40 “Algoritmo – Fuerza Bruta” (Elaboración Propia).....	69

## 1. Catálogo de actores

- **Restaurante:** este tipo de usuario cumple con las características de representar a un restaurante para poder incluirlo en la guía gastronómica. Este se encargará de registrar toda la información del restaurante que sea necesaria para poder publicarlo en el sistema, administrará su catálogo de platos, podrá consultar y reportar críticas que otros usuarios hayan realizado hacia él y podrá consultar sus quejas (críticas reportadas) para saber si han sido atendidas por los administradores.
- **Usuario Público:** este actor es el único que no necesita encontrarse registrado en el sistema. Sus funciones son las de consultar toda la información que desee acerca de los restaurantes registrados en el sistema. Este gozará de poder conocer las distancias con respecto a los restaurantes, su calificación general, y toda aquella información pública.
- **Consumidor:** este actor es cualquier persona natural que desee poder calificar y criticar los restaurantes publicados en el sistema. Este cuenta con las mismas funcionalidades que el usuario público a excepción de las de calificar a un restaurante y poseer un perfil, en el cual puede consultar todas las críticas que ha realizado, ya sean aprobadas, rechazadas o pendientes de verificación.
- **Administrador:** este actor se encarga de realizar la administración del sistema. Por un lado, administra los maestros de servicios, tipos de local, tipos de cocina, medios de pago con los que los usuarios restaurante podrán registrar sus locales. Por otro lado, se encarga de administrar el flujo de las críticas y quejas que registran los comensales y restaurantes. Finalmente, realiza la configuración adecuada del mecanismo automático que analiza el contenido textual de las críticas

## 2. Casos de uso

En la figura 1, se muestra el paquete de seguridad que es utilizado por todos los actores que ya pertenecen al sistema. Este consta de las funcionalidades básicas de todo módulo de seguridad, la autenticación a través de un inicio de sesión, la identificación del rol del usuario, el cambio de contraseña y el desbloqueo de cuentas.

□

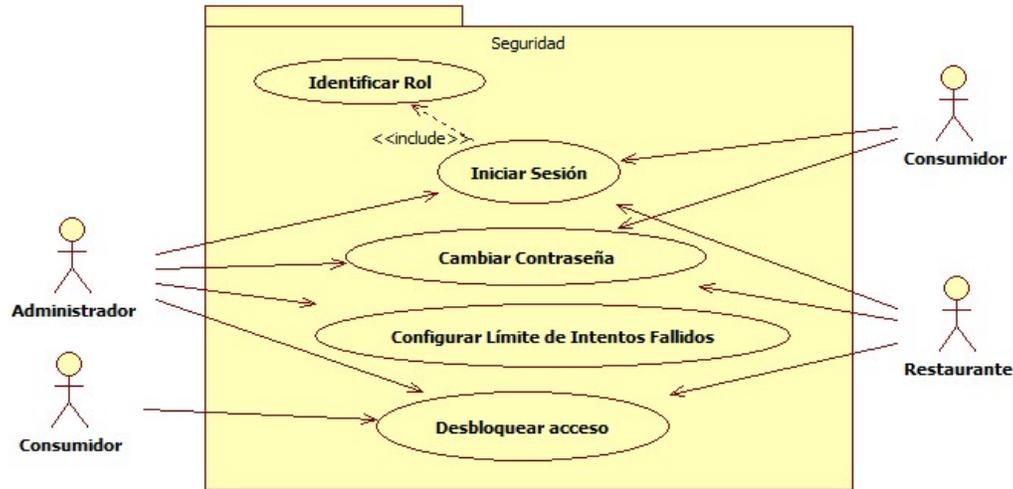


Figura 1 “Seguridad – Casos de Uso” (Elaboración Propia)

En la figura 2, se muestra el paquete de restaurantes, este se enfoca en solo el actor restaurante y todas las funcionalidades que el utiliza. En el paquete se puede observar que se cuentan con las opciones de enviar la solicitud de registro, la administración de su información, la administración de su catálogo de platos, la consulta de sus críticas y la posibilidad de reportar estas y la consulta de sus quejas registradas

En la figura 3, se muestra el paquete de la administración del sistema, similar al paquete anterior, este se centra en todas aquellas funcionalidades a las que tiene acceso el usuario administrador. Cabe resaltar que la configuración de la gestión de las críticas automatizada consiste en la configuración de las entradas del algoritmo expuesto en los objetivos específicos para analizar los textos de las críticas, y así decidir si debe ser aceptado o no.

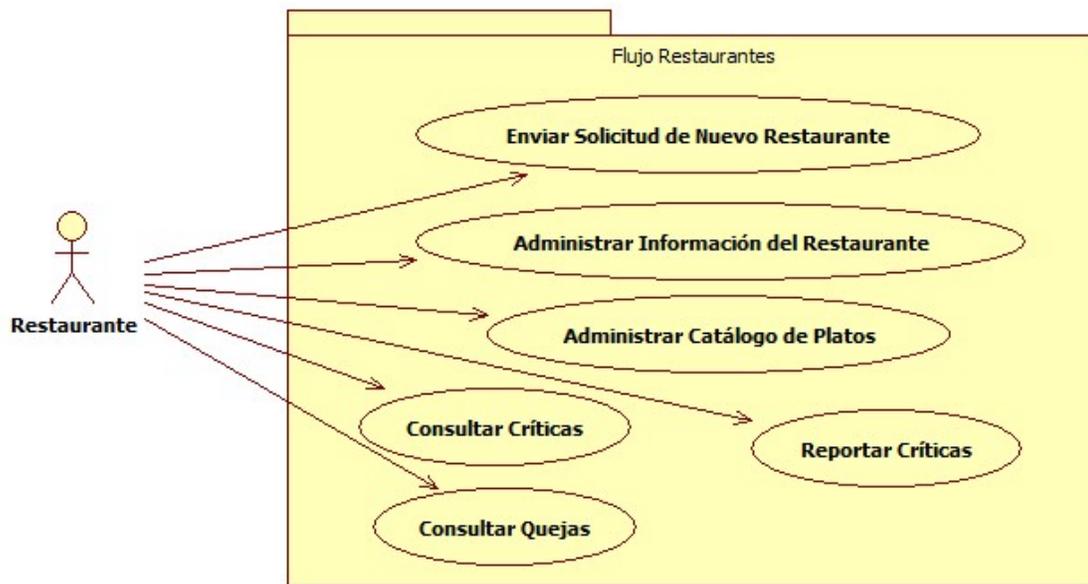


Figura 2 “Restaurantes – Casos de Uso” (Elaboración Propia)

Por último, en la figura 4, el flujo de consumidores muestra todas aquellas funcionalidades a las que tienen acceso los comensales y los usuarios públicos. Cabe resaltar que de la acción de consultar restaurantes se encuentra una búsqueda avanzada por filtros, para que el comensal pueda ubicar lo que necesita con mayor rapidez, y una búsqueda normal por nombre. Asimismo, esta búsqueda la acción de realizar esta búsqueda ejecuta de lado del servidor el algoritmo de ordenamiento expuesto en los objetivos específicos, el cual tomará las distancias entre usuario y restaurantes como entrada.

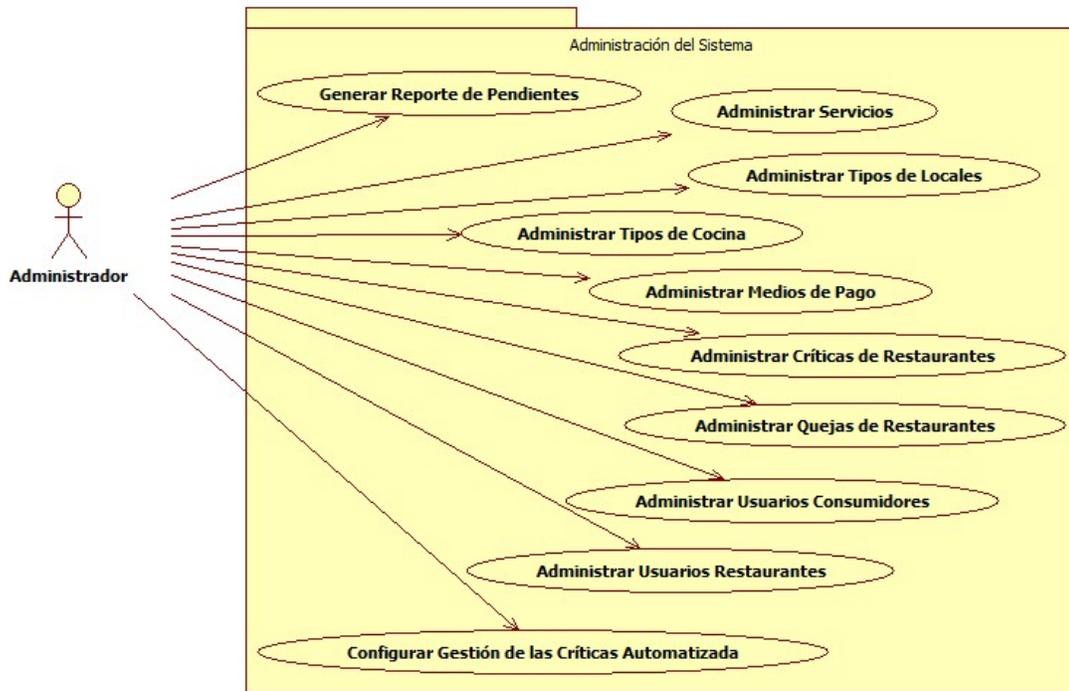


Figura 3 “Administración – Casos de Uso” (Elaboración Propia)

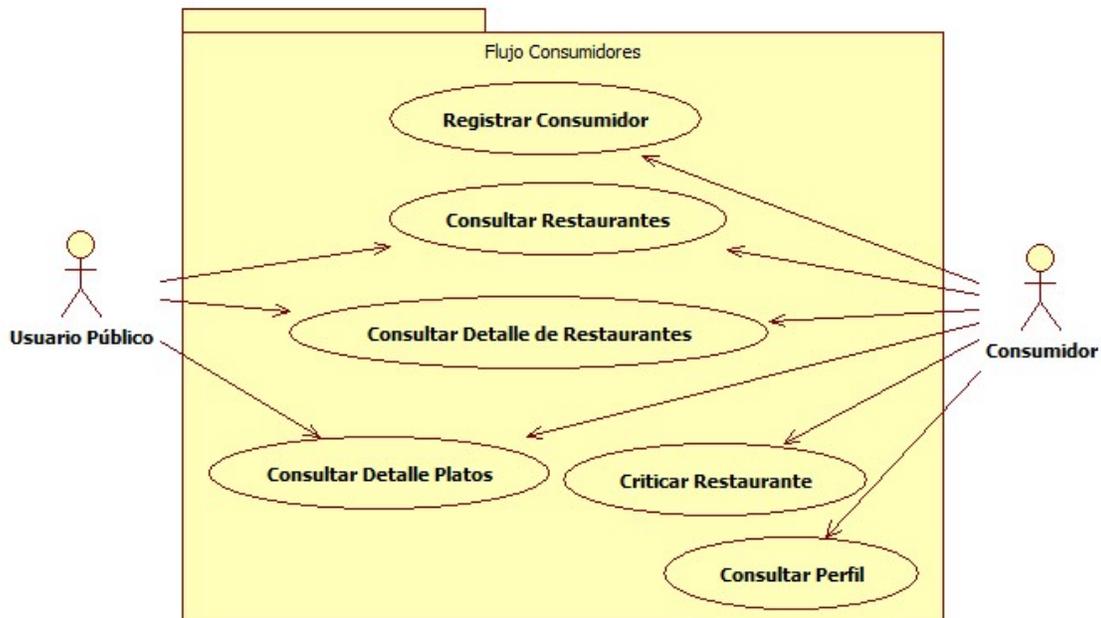


Figura 4 “Consumidores – Casos de Uso” (Elaboración Propia)

Los códigos usados para los casos de uso se muestran en la siguiente tabla:

	Caso de Uso
CU-SEG-01	Identificar Rol
CU-SEG-02	Iniciar Sesión
CU-SEG-03	Cambiar Contraseña
CU-SEG-04	Desbloquear Acceso
CU-SEG-05	Configurar Límite de Intentos Fallidos
CU-COM-01	Registrar Consumidor
CU-COM-02	Consultar Restaurantes
CU-COM-03	Consultar Detalle de Restaurantes
CU-COM-04	Consultar Detalle de Platos
CU-COM-05	Criticar Restaurante
CU-COM-06	Consultar Perfil
CU-REST-01	Enviar Solicitud de Nuevo Restaurante
CU-REST-02	Administrar Información del Restaurante
CU-REST-03	Administrar Catálogo de Platos
CU-REST-04	Consultar Críticas
CU-REST-05	Consultar Quejas
CU-REST-06	Reportar Críticas
CU-ADMIN-01	Generar Reporte de Pendientes
CU-ADMIN-02	Administrar Servicios
CU-ADMIN-03	Administrar Tipos de Locales
CU-ADMIN-04	Administrar Tipos de Cocina
CU-ADMIN-05	Administrar Medios de Pago
CU-ADMIN-06	Administrar Críticas de Restaurantes
CU-ADMIN-07	Administrar Quejas de Restaurantes
CU-ADMIN-08	Administrar Usuarios Consumidores
CU-ADMIN-09	Administrar Usuarios Restaurantes
CU-ADMIN-10	Configurar Gestión de las Críticas Automatizada

### 3. Especificación de casos de uso

#### 3.1 Paquete de Seguridad

Identificar Rol		
Datos	<b>Id</b>	CU-SEG-001
	<b>Descripción</b>	Este caso de uso permite a los distintos actores del sistema encontrar el rol en el que se encuentran.
	<b>Referencia a lista de requerimientos</b>	FUN-01.
	<b>Paquete</b>	Seguridad
	<b>Actores</b>	Administrador, Consumidor, Restaurante.
<b>Pre-condición</b>		El actor debe encontrarse en el formulario de autenticación del sistema.
<b>Flujo principal: “Identificar Rol”</b>		
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>	
1. El actor debe ingresar el nombre de usuario y la contraseña.	2. El sistema valida los campos ingresados y reconoce al usuario y su rol. Si el usuario es un comensal, mostrará la misma interfaz que cuando no se había conectado aún; si el usuario es un restaurante, mostrará un menú de opciones: <i>Administrar información, catálogo de platos, críticas y quejas</i> ; si el usuario es un administrador, las opciones en el menú serán <i>servicios, tipos de locales, tipos de cocinas, medios de</i>	

	<i>pago, críticas, quejas, comensales, restaurantes, configuración.</i>
<b>Post-condición Flujo principal</b>	El sistema muestra la interfaz correspondiente al usuario según su rol.
<b>Flujo alternativo 1: “Ingresar Datos Incorrectos”</b>	
1. Acción del punto 1 del flujo principal.	2. El sistema valida los campos ingresados y muestra el mensaje “Usuario y/o contraseña inválidos” y solicita nuevamente los datos.
3. El actor repite la acción del punto 1 hasta que ingrese correctamente los datos.	4. El sistema muestra la interfaz como se indicó en el punto 2 del flujo principal.
<b>Post-condición Flujo alternativo 1:</b>	El sistema muestra la interfaz correspondiente al usuario según su rol.
<b>Flujo alternativo 2: “Bloquear Usuario”</b>	
1. El actor repite por tercera vez consecutiva la acción del punto 1 del flujo alternativo 1.	2. El sistema bloquea el acceso al usuario.
<b>Post-condición Flujo alternativo 2:</b>	El usuario queda deshabilitado de ingresar al sistema.

Iniciar Sesión		
Datos	<b>Id</b>	CU-SEG-002
	<b>Descripción</b>	Este caso de uso permite a los distintos actores del sistema poder ingresar a este a través de una autenticación.
	<b>Referencia a lista de requerimientos</b>	FUN-40, FUN-41.

	<b>Paquete</b>	Seguridad
	<b>Actores</b>	Administrador, Consumidor, Restaurante.
<b>Pre-condición</b>		
		El actor debe encontrarse en la página de bienvenida del sistema.
<b>Flujo principal: “Iniciar Sesión”</b>		
	<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
	1. El actor debe ingresar a la opción de <i>Iniciar Sesión</i> .	2. El sistema muestra un formulario con los campos de nombre de usuario y contraseña.
	3. El actor sigue la secuencia indicada en el caso de uso CU-SEG-001.	4. El sistema muestra la interfaz correspondiente al usuario según su rol.
<b>Post-condición Flujo principal</b>		El usuario ha iniciado sesión.

<b>Cambiar contraseña</b>		
Datos	<b>Id</b>	CU-SEG-003
	<b>Descripción</b>	Este caso de uso permite a los distintos actores registrados en el sistema cambiar su contraseña.
	<b>Referencia a lista de requerimientos</b>	FUN-43
	<b>Paquete</b>	Seguridad
	<b>Actores</b>	Administrador Secundario, Consumidor, Restaurante.
<b>Pre-condición</b>		
		El actor debe encontrarse en el formulario de autenticación del sistema
<b>Flujo principal: “Cambiar contraseña”</b>		

Acción del actor	Respuesta del Sistema
1. El actor debe seleccionar la opción <i>cambiar contraseña..</i>	2. El sistema muestra un campo en el que se debe ingresar el nombre de usuario, su contraseña actual, la nueva contraseña y una opción <i>Registrar.</i>
3. El actor ingresa los datos correspondientes y selecciona la opción <i>Registrar.</i>	4. El sistema valida la contraseña y muestra un mensaje indicando lo siguiente: <i>su contraseña ha sido modificada.</i>
<b>Post-condición Flujo principal</b>	El usuario inicia sesión.

Desbloquear acceso		
Datos	<b>Id</b>	CU-SEG-004
	<b>Descripción</b>	Este caso de uso permite a los distintos actores del sistema desbloquear su usuario en caso hayan este se encuentre en dicho estado..
	<b>Referencia a lista de requerimientos</b>	FUN-42.
	<b>Paquete</b>	Seguridad
	<b>Actores</b>	Administrador, Consumidor, Restaurante.
<b>Pre-condición</b>		El actor debe encontrarse en el formulario de autenticación del sistema.
Flujo alternativo 3: “Desbloquear Usuario”		
1. El actor selecciona la opción <i>Desbloquear usuario</i>	2. El sistema solicita el correo electrónico con el que se registró el usuario y la opción <i>Enviar.</i>	

3. El actor ingresa su correo electrónico y la opción enviar.	5. El sistema envía una nueva contraseña al correo electrónico del usuario.
4. El usuario ingresa con la nueva contraseña que se le ha enviado.	5. El sistema valida los datos y desbloquea el ingreso al usuario.
<b>Post-condición Flujo alterno 3:</b>	El usuario queda habilitado nuevamente para ingresa al sistema.

### 3.2 Paquete de Administración

Administrar Críticas de Restaurantes		
Datos	<b>Id</b>	CU-ADMIN-006
	<b>Descripción</b>	Consiste en administrar las críticas realizadas por los consumidores hacia los restaurantes.
	<b>Referencia a lista de requerimientos</b>	FUN-10, FUN-11.
	<b>Paquete</b>	Administración del Sistema
	<b>Actores</b>	Administrador
<b>Pre-condición</b>		El actor debe haber iniciado sesión.
<b>Flujo principal: “Aceptar Críticas”</b>		
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>	
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 registro, su <i>estado</i> , el cual será <i>Pendiente</i> para los que aún no han sido revisados, y la	

	opción <i>Editar</i> .
3. El actor seleccionará la opción <i>Editar</i> correspondiente a cualquiera de las críticas.	4. El sistema mostrará los siguientes datos con respecto a la crítica: su <i>calificación (comida, servicio, ambiente y precio)</i> , el <i>comentario</i> , el <i>nombre del comensal</i> que realizó la crítica, el <i>nombre del restaurante</i> al cual se criticó y su <i>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> .
5. El actor asignará el estado <i>Aprobado</i> y seleccionará la opción <i>Guardar cambios</i> .	6. El sistema muestra el siguiente mensaje: <i>La crítica se ha sido aprobada en el sistema</i> . Asimismo, el sistema promedia la calificación de esta crítica con la calificación general del restaurante.
<b>Post-condición Flujo principal</b>	La crítica se encuentra en estado aprobado en el sistema.
<b>Flujo alternativo 1: “Rechazar Críticas”</b>	
1. En el punto 5 del flujo principal, el actor asigna el estado <i>Desaprobado</i> .	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> .
3. El actor redacta las razones correspondientes en el cuadro y selecciona la opción <i>Enviar</i> .	4. El sistema muestra el siguiente mensaje: <i>Se ha enviado un correo al comensal indicándole las razones</i>

	<i>por las que no pudo registrarse su crítica en el sistema. Además, se muestra la opción <i>Aceptar</i>.</i>
5. El actor selecciona la opción <i>Aceptar</i>	6. Se muestra el listado inicial del punto 2 del flujo principal.
<b>Post-condición Flujo alterno 1:</b>	Las críticas quedará registrado en un estado de desaprobada en el sistema.

Administrar Quejas de Restaurantes		
Datos	<b>Id</b>	CU-ADMIN-007
	<b>Descripción</b>	Consiste en administrar las quejas realizadas por los restaurantes con respecto a críticas que consideren no adecuadas para mostrarse públicamente.
	<b>Referencia a lista de requerimientos</b>	FUN-12, FUN-13, FUN-13.
	<b>Paquete</b>	Administración del Sistema
	<b>Actores</b>	Administrador
<b>Pre-condición</b>		El actor debe haber iniciado sesión.
Flujo principal: "Aceptar Queja"		
Acción del actor		Respuesta del Sistema
1. El actor debe seleccionar la opción <i>Quejas</i> .		2. El sistema mostrará un listado con el <i>código de la queja</i> , el <i>nombre del restaurante</i> , la <i>fecha</i> en que se registro, su <i>estado</i> , el cual será <i>Pendiente</i> para los que aún no han sido revisados, y la opción <i>Editar</i> .
3. El actor seleccionará la opción		4. El sistema mostrará los siguientes

<p><i>Editar</i> correspondiente a cualquiera de las críticas.</p>	<p>datos con respecto a la queja: el <i>nombre del restaurante</i> que se quejó, la <i>razón</i> por la cual se quejó, el <i>nombre del comensal</i> que realizó la crítica, la <i>calificación (comida, servicio, ambiente y precio)</i> que colocó, el <i>comentario</i> y el <i>estado</i> de la queja. Este último será el único editable para poder aceptar o rechazar la queja. Por último, se muestra la opción <i>Guardar cambios</i>.</p>
<p>5. El actor asignará el estado <i>Aprobado</i> y seleccionará la opción <i>Guardar cambios</i>.</p>	<p>6. El sistema muestra el siguiente mensaje: <i>La queja se ha sido aprobada en el sistema</i>. Asimismo, el sistema desaprueba la crítica asociada a la queja y la retira del cálculo de la calificación general.</p>
<p><b>Post-condición Flujo principal</b></p>	<p>La queja se encuentra en estado aprobado en el sistema y su crítica asociada en estado desaprobado.</p>
<p style="text-align: center;"><b>Flujo alternativo 1: “Rechazar Queja”</b></p>	
<p>7. En el punto 5 del flujo principal, el actor asigna el estado <i>Desaprobado</i>.</p>	<p>8. 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 queja. Además, se muestra la opción <i>Enviar</i> y <i>Cancelar</i>.</p>
<p>9. El actor redacta las razones correspondientes en el cuadro y selecciona la opción <i>Enviar</i>.</p>	<p>10. El sistema muestra el siguiente mensaje: <i>Se ha enviado un correo al restaurante indicándole las razones</i></p>

	<i>por las que no pudo registrarse su queja en el sistema. Además, se muestra la opción <i>Aceptar</i>.</i>
11. El actor selecciona la opción <i>Aceptar</i>	12. Se muestra el listado inicial del punto 2 del flujo principal.
<b>Post-condición Flujo alternativo 1:</b>	La queja quedará registrada en un estado de desaprobado.

Administrar Usuarios Consumidores		
Datos	<b>Id</b>	CU-ADMIN-008
	<b>Descripción</b>	Este caso de uso permite, al administrador del sistema, administrar a los usuarios comensales.
	<b>Referencia a lista de requerimientos</b>	FUN-04.
	<b>Paquete</b>	Administración del Sistema
	<b>Actores</b>	Administrador
<b>Pre-condición</b>	El actor debe haber iniciado sesión.	
Flujo principal: “Desactivar Comensal”		
Acción del actor	Respuesta del Sistema	
1. El actor debe ingresar a la opción <i>Comensales</i> .	2. El sistema mostrará un listado con el <i>código del comensal</i> , el <i>nombre del usuario</i> , los <i>nombres de la persona</i> , los <i>apellidos de la persona</i> , la <i>fecha</i> en que se registró, su <i>estado</i> , el cual será <i>Activado</i> , y la opción <i>Editar</i> .	
3. El actor selecciona la opción <i>Editar</i> de cualquier comensal.	4. El sistema mostrará el detalle del comensal seleccionado con los	

	siguientes datos: <i>nombre de usuario, nombres de la persona, apellidos de la persona, la cantidad de críticas y el estado</i> . Este último será el único editable para poder desactivar al usuario. Por último, se muestra la opción <i>Guardar cambios</i>
5. El actor selecciona el estado <i>Desactivado</i> .	6. El sistema muestra el siguiente mensaje: <i>El comensal ha sido desactivado</i> .
<b>Post-condición Flujo principal</b>	El comensal queda en un estado desactivado.
<b>Flujo alternativo 1: "Activar Comensal"</b>	
1. El actor realiza el mismo paso 1 del flujo principal.	2. El sistema muestra lo mismo del paso 2 del flujo principal, pero con estado <i>Desactivado</i> .
3. El actor realiza el mismo paso 3 del flujo principal hasta el paso 5, pero elige el estado <i>Activado</i> .	4. El sistema muestra un listado de usuarios con sus datos, "Cantidad de Críticas", "Fecha de Última Crítica", "Porcentaje de Aceptación de Críticas" y "Cantidad de Críticas Rechazadas".
5. El actor selecciona uno de los usuarios listados.	6. El sistema muestra el siguiente mensaje: <i>El comensal ha sido activado</i> .
<b>Post-condición Flujo alternativo 1:</b>	El comensal queda en un estado activado.

Administrar Usuarios Restaurantes		
Datos	<b>Id</b>	CU-ADMIN-009
	<b>Descripción</b>	Consiste en administrar las solicitudes por parte de restaurantes

		para poder hacer uso del sistema.
<b>Referencia a lista de requerimientos</b>	<b>a</b>	FUN-05, FUN-15.
<b>Paquete</b>		Administración del Sistema
<b>Actores</b>		Administrador.
<b>Pre-condición</b>		El actor debe haber iniciado sesión.
<b>Flujo principal: “Aceptar Solicitudes”</b>		
<b>Acción del actor</b>		<b>Respuesta del Sistema</b>
7. El actor debe seleccionar la opción <i>Restaurantes</i> .		8. El sistema mostrará un listado con el <i>código del restaurante</i> , el <i>nombre del usuario restaurante</i> , el <i>nombre del restaurante</i> , la <i>fecha</i> en que se registro, su <i>estado</i> , el cual será <i>Pendiente</i> para los que recién han enviado su solicitud, y la opción <i>Editar</i> .
9. El actor seleccionará la opción <i>Editar</i> correspondiente a cualquiera de los restaurantes.		10. El sistema mostrará el detalle de la solicitud con los siguientes datos: <i>nombre de usuario</i> , <i>nombre del restaurante</i> , <i>número telefónico</i> , <i>zona</i> , <i>dirección</i> y el <i>estado</i> . Este último será el único editable para poder aceptar o rechazar la solicitud. Por último, se muestra la opción <i>Guardar cambios</i> .
11. El actor asignará el estado <i>Activado</i> y seleccionará la opción <i>Guardar cambios</i> .		12. El sistema muestra el siguiente mensaje: <i>El restaurante se ha activado en el sistema</i> .

<b>Post-condición Flujo principal</b>	El restaurante se encuentra activado en el sistema.
<b>Flujo alternativo 1: “Rechazar Solicitudes”</b>	
1. En el punto 5 del flujo principal, el actor asigna el estado <i>Desactivado</i> .	2. El sistema muestra un cuadro con el título de <i>Razones</i> , en el cual se redacta las razones por las que fue rechazado el restaurante. Además, se muestra la opción <i>Enviar</i> y <i>Cancelar</i> .
3. El actor redacta las razones correspondientes en el cuadro y selecciona la opción <i>Enviar</i> .	4. El sistema muestra el siguiente mensaje: <i>Se ha enviado un correo al restaurante indicándole las razones por las que no pudo registrarse en el sistema</i> . Además, se muestra la opción <i>Aceptar</i> .
5. El actor selecciona la opción <i>Aceptar</i>	6. Se muestra el listado inicial del punto 2 del flujo principal.
<b>Post-condición Flujo alternativo 1:</b>	El restaurante se encuentra en estado desactivado en el sistema.

Configurar Gestión de las Críticas Automatizada		
Datos	<b>Id</b>	CU-ADMIN-010
	<b>Descripción</b>	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.
	<b>Referencia a lista de requerimientos</b>	FUN-17, FUN-18.
	<b>Paquete</b>	Administración del Sistema
	<b>Actores</b>	Administrador

<b>Pre-condición</b>	El actor debe haber iniciado sesión.
<b>Flujo principal: “Activar gestión automatizada”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
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.
<b>Post-condición Flujo principal</b>	El mecanismo de administración de críticas automatizado se encuentra activado en el sistema.
<b>Flujo alterno 1: “Registrar palabra”</b>	
13. En el punto 3 del flujo principal, selecciona la opción <i>Registrar nueva palabra</i> .	14. El sistema muestra una campo editable en el que se ingresará la <i>palabra</i> a registrar y la opción <i>Registrar</i> .

15. El actor redacta la nueva palabra y seleccionará la opción <i>Registrar</i> .	16. El sistema muestra el siguiente mensaje: <i>Se ha registrado la palabra exitosamente</i> . Además, muestra la opción <i>Aceptar</i> .
17. El actor selecciona la opción <i>Aceptar</i> .	18. Se muestra el listado inicial del punto 2 del flujo principal.
<b>Post-condición Flujo alterno 1:</b>	La palabra ha quedado registrada en el sistema.

### 3.3 Paquete de Restaurantes

Enviar Solicitud de Nuevo Restaurante		
Datos	<b>Id</b>	CU-REST-001
	<b>Descripción</b>	Consiste en que los restaurantes puedan enviar solicitudes con información sobre ellos para que puedan formar parte del sistema y, así poder publicar su información y administrarla.
	<b>Referencia a lista de requerimientos</b>	FUN-19.
	<b>Paquete</b>	Flujo Restaurantes
	<b>Actores</b>	Restaurante
<b>Pre-condición</b>	El actor aún no se encuentra registrado en el sistema y se encuentra en la página principal del sistema.	
Flujo principal: “Enviar Solicitud”		
Acción del actor	Respuesta del Sistema	
1. El actor debe seleccionar la opción <i>Agregar Restaurante</i> .	2. El sistema mostrará distintas secciones en las que se deberá colocar la información sobre el	

	<p>restaurante. Primero se encuentran los <i>Datos Generales</i> con los siguientes campos: <i>nombre del restaurante, ciudad, zona, dirección, teléfono, fecha de apertura</i>. Luego, se encuentra el <i>tipo de cocina, tipo de lugar, medios de pago, servicios, ubicación en el mapa y foto del restaurante</i>. Además, se mostrará la opción <i>siguiente</i>.</p>
<p>3. El actor llenará los campos mencionados en el punto 2 con la información correspondiente y seleccionará la opción <i>Enviar Restaurante</i>.</p>	<p>4. El sistema verifica que los campos hayan sido llenados y mostrará otra interfaz en la que se soliciten los siguientes datos: <i>nombre de usuario, correo electrónico, contraseña, la contraseña nuevamente y los nombres y apellidos de la persona</i>. Finalmente, se mostrará una opción <i>Enviar Solicitud</i>.</p>
<p>5. El actor llenará los campos mencionados en el punto 4 con la información correspondiente y seleccionará la opción <i>Enviar Solicitud</i>.</p>	<p>6. El sistema muestra un mensaje indicando lo siguiente: <i>Su solicitud ha sido enviada exitosamente</i>.</p>
<p><b>Post-condición Flujo principal</b></p>	<p>Una solicitud de nuevo restaurante queda registrada en el sistema.</p>

Administrar Información del Restaurante		
Datos	<b>Id</b>	CU-REST-002
	<b>Descripción</b>	Consiste en que el usuario restaurante pueda administrar la información que publicará en el sistema.

<b>Referencia a lista de requerimientos</b>	FUN-02, FUN-20, FUN-21, FUN-22, FUN-23, FUN-24.
<b>Paquete</b>	Flujo Restaurantes
<b>Actores</b>	Restaurante
<b>Pre-condición</b>	
	El actor ha iniciado sesión.
<b>Flujo principal: “Administrar Información”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
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.
<b>Post-condición Flujo principal</b>	Se actualizaron los datos del restaurante en el sistema.

Consultar Críticas		
Datos	<b>Id</b>	CU-REST-004
	<b>Descripción</b>	Consiste en que el usuario restaurante pueda consultar su calificación global y todas las críticas aportadas por los comensales.
	<b>Referencia a lista de requerimientos</b>	FUN-27.
	<b>Paquete</b>	Flujo Restaurantes
	<b>Actores</b>	Restaurante
<b>Pre-condición</b>		El actor ha iniciado sesión.
Flujo principal: “Consultar Críticas”		
Acción del actor		Respuesta del Sistema
1. El actor debe seleccionar la opción <i>Críticas</i> .		2. El sistema mostrará en una primera sección la <i>calificación general</i> del restaurante ( <i>Comida, Servicio, Ambiente, Precio, Votos</i> ). Por otro lado, en una segunda sección, mostrará todas las críticas registradas y aprobadas. Por cada una de ellas, se mostrará el <i>nombre del comensal</i> , la <i>calificación (Comida, Servicio, Ambiente, Precio)</i> , el <i>comentario</i> y una opción <i>Reportar</i> .
<b>Post-condición Flujo principal</b>		Se mostraron las críticas del restaurante.

Reportar Críticas		
D	<b>Id</b>	CU-REST-005

	<b>Descripción</b>	Consiste en que el usuario restaurante pueda reportar una de las críticas.
	<b>Referencia a lista de requerimientos</b>	FUN-28.
	<b>Paquete</b>	Flujo Restaurantes
	<b>Actores</b>	Restaurante
<b>Pre-condición</b>		El actor ha iniciado sesión.
<b>Flujo principal: “Administrar Información”</b>		
<b>Acción del actor</b>		<b>Respuesta del Sistema</b>
1. El actor sigue el flujo del caso de uso CU-REST-004.		2. El sistema mostrará el <i>nombre del comensal</i> , la <i>calificación (comida, servicio, ambiente y precio)</i> y el <i>comentario</i> . A continuación, el sistema muestra un campo editable en el que se ingresará la <i>razón</i> de la queja. Finalmente, se muestra la opción <i>Registrar</i> .
<b>Post-condición Flujo principal</b>		Se registró la queja en el sistema.

### 3.4 Paquete de Consumidores

Consultar Restaurantes		
Datos	<b>Id</b>	CU-COM-002
	<b>Descripción</b>	Este caso de uso muestra cómo se realizan las búsquedas de restaurantes en el sistema.
	<b>Referencia a lista de</b>	FUN-33, FUN-34, FUN-35.

<b>requerimientos</b>	
<b>Paquete</b>	Flujo Consumidores
<b>Actores</b>	Consumidor, Usuario Público.
<b>Pre-condición</b>	El actor del debe encontrarse en la página inicial sistema.
<b>Flujo principal: “Búsqueda Normal”</b>	
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>
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>
<b>Post-condición Flujo principal</b>	Se mostró un listado de restaurantes.
<b>Flujo alterno 1: “Búsqueda Avanzada”</b>	
1. El actor selecciona la opción <i>Búsqueda Avanzada</i> .	2. El sistema muestra cinco criterios distintos para la búsqueda: <i>Tipos de Cocina, Zona, Tipo de Local, Rango de Precios, Servicios, Medios de Pago</i> y Distancia Máxima. 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 criterios que desee según sus necesidades y seleccionar la opción	4. El sistema mostrará el mismo resultado del punto 2 en el flujo principal.

<i>Buscar.</i>	
<b>Post-condición Flujo alterno 1:</b>	Se mostró un listado de restaurantes.

Consultar Detalle de Restaurantes		
Datos	<b>Id</b>	CU-COM-003
	<b>Descripción</b>	Este caso de uso describe el resultado de consultar el detalle de cualquier restaurante.
	<b>Referencia a lista de requerimientos</b>	FUN-36, FUN-37.
	<b>Paquete</b>	Flujo Consumidores
	<b>Actores</b>	Consumidor, Usuario Público.
<b>Pre-condición</b>	El actor del debe encontrarse en la página inicial sistema.	
<b>Flujo principal: “Consultar Detalle Restaurante”</b>		
<b>Acción del actor</b>	<b>Respuesta del Sistema</b>	
1. El actor consulta los restaurantes siguiendo cualquiera de los flujos del caso de uso CU-COM-002.	2. El sistema muestra un listado de restaurantes.	
3. El actor selecciona el nombre de uno de los resultados de la lista.	4. El sistema mostrará el nombre del restaurante. Luego, en una primera sección denominada <i>Datos Generales</i> , mostrará el <i>distrito</i> , la <i>dirección</i> , el <i>teléfono</i> , el <i>tipo de local</i> , el <i>tipo de cocina</i> , un listado de sus <i>servicios</i> , un listado de sus <i>medios de pago</i> y su <i>calificación general</i> . En otra sección, se muestra la <i>ubicación</i> del restaurante en un mapa y una imagen del local.	

	Además, mostrará un catálogo de platos, el cual se podrá navegar hasta el detalle de cada plato. Por último, se muestra una opción <i>Calificar Restaurante</i> y todas las calificaciones en detalle de cada usuario.
<b>Post-condición Flujo principal</b>	El sistema mostró el detalle de todo el restaurante

Criticar Restaurante		
Datos	<b>Id</b>	CU-COM-005
	<b>Descripción</b>	Este caso de uso muestra el flujo de cómo el cliente califica a un restaurante.
	<b>Referencia a lista de requerimientos</b>	FUN-03, FUN-31, FUN-32.
	<b>Paquete</b>	Flujo Consumidores
	<b>Actores</b>	Consumidor.
<b>Pre-condición</b>		El actor debe haber iniciado sesión.
<b>Flujo principal: “Calificar Restaurante”</b>		
<b>Acción del actor</b>		<b>Respuesta del Sistema</b>
3. El actor consulta el detalle de un restaurante siguiendo el flujo del caso de uso CU-COM-003.		4. El sistema mostrará el detalle de todo el restaurante.
5. El actor selecciona la opción <i>Calificar Restaurante</i> .		6. El sistema mostrará el <i>nombre del restaurante</i> . Asimismo, mostrará cuatro criterios de evaluación: <i>Comida, Servicio, Ambiente</i> y

	<p><i>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> 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>.</p>
<p>7. El actor llenará los campos mencionados en el punto anterior y seleccionará la opción <i>Calificar</i>.</p>	<p>8. 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></p>
<p><b>Post-condición Flujo principal</b></p>	<p>Se registró una calificación en el sistema.</p>
<p><b>Flujo alternativo 1: “Criterios Vacíos”</b></p>	
<p>6. En el punto 5 del flujo principal, el actor no llena todos los criterios de calificación y selecciona la opción <i>Calificar</i>.</p>	<p>7. El sistema valida los criterios y, para los que estén vacíos, mostrará el mensaje <i>Por favor indique un valor para este criterio.</i></p>
<p>8. El actor volverá al punto 5 del flujo principal hasta que llene correctamente todos los criterios y seleccione la opción <i>Calificar</i>.</p>	<p>9. 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></p>

<b>Post-condición Flujo alternativo 1:</b>	Se registró una calificación en el sistema.
<b>Flujo alternativo 2: “Comentarios incompletos”</b>	
3. En el punto 5 del flujo principal, el actor llena los criterios, coloca un comentario con menos de 100 caracteres y selecciona la opción <i>Votar</i> .	4. El sistema valida los criterios y el comentarios, y muestra un mensaje <i>La reseña debe tener más de 100 caracteres</i> .
5. 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> .	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> .
<b>Post-condición Flujo alternativo 2:</b>	Se registró una calificación en el sistema.

Consultar Perfil		
Datos	<b>Id</b>	CU-COM-006
	<b>Descripción</b>	En este caso de uso se consulta el perfil del comensal, el cual mostrará sus críticas registradas en el sistema.
	<b>Referencia a lista de requerimientos</b>	FUN-39.
	<b>Paquete</b>	Flujo Consumidores
	<b>Actores</b>	Consumidor.
<b>Pre-condición</b>		El actor debe haber iniciado sesión.
<b>Flujo principal: “Consultar Perfil”</b>		
<b>Acción del actor</b>		<b>Respuesta del Sistema</b>



N - 0 2																				
F U N - 0 3																				
F U N - 0 4																				
F U N - 0 5																				
F U N - 0 6																				
F U N - 0 7																				
F U N - 0 8																				
F U N - 0 9																				











id (PK)	int	Identificador del registro del usuario.
userName	varchar(20)	Nombre del usuario con el que se autenticará.
password	varchar(50)	Contraseña del usuario con la que se autenticará.
email	varchar(50)	Correo electrónico del usuario para poder contactarlo.
estadoId (FK)	int	Identificador de la tabla Estado para reconocer si el usuario está activado o desactivado.
fecha	date	Fecha en la que se registró el usuario.
rolId (FK)	int	Identificar de la tabla Rol para reconocer a qué rol pertenece el usuario.

### Estado

Tabla que almacena la información de los distintos estados que pueden tener los registros de las tablas usuario, crítica y queja.

Columna	Tipo	Descripción
id (PK)	int	Identificador del registro del estado.
descripción	varchar(30)	Descripción del estado.
tipo	varchar(30)	Descripción que se utiliza como identificador para reconocer si se refiere a un estado de la tabla usuario, crítica o queja.

### Comensal

Tabla que almacena la información de los comensales registrados en el sistema.

Columna	Tipo	Descripción
id (PK)	int	Identificador del registro del comensal.
usuarioId (FK)	int	Identificador del usuario al que está asociado para que pueda autenticarse.
nombres	varchar(50)	Los nombres del comensal.
apellidos	varchar(50)	Los apellidos del comensal.
cantidadCriticas	int	Permite llevar la cuenta de la cantidad de críticas

		registradas en el sistema por cada comensal.
--	--	----------------------------------------------

Crítica		
Tabla que almacena la información de las críticas registradas en el sistema.		
Columna	Tipo	Descripción
id (PK)	int	Identificador del registro de la crítica.
valorComida	int	Valor que se le asigna por la calidad de la comida, el cual varía de 1 al 5 haciendo referencia de peor a mejor servicio.
valorServicio	int	Valor que se le asigna por el servicio de atención, el cual varía de 1 al 5 haciendo referencia de peor a mejor servicio
valorAmbiente	int	Valor que se le asigna por el ambiente, el cual varía de 1 al 5 haciendo referencia de peor a mejor servicio
precio	int	Precio promedio por la adquisición de una entrada más un plato de fondo + una bebida.
comentario	varchar(250)	Contenido del comentario que forma parte de la crítica.
comensalId (FK)	int	Identificador que asocia la crítica con un comensal el cual realizó la crítica.
restauranteId (FK)	int	Identificador que asocia la crítica con un restaurante hacia el que va dirigido.
estadoId (FK)	int	Identificador que asocia la crítica con un estado para reconocer si está aprobada, desaprobada o pendiente.
fecha	date	Permite reconocer cuándo fue registrada la crítica

**Queja**

Tabla que almacena la información de las quejas (críticas reportadas) registradas en el sistema.		
Columna	Tipo	Descripción
id (PK)	int	Identificador del registro de la queja.
descripcion	varchar(250)	Razón por la que se está reportando la crítica asociada
criticalId (FK)	int	Identificador que asocia la queja con una crítica.
estadoId (FK)	int	Identificador que asocia la queja con un estado para reconocer si está aprobada, desaprobada o pendiente
fecha	date	Permite reconocer cuándo fue registrada la queja.

<b>Opcion</b>		
Tabla que almacena los patrones a analizar en los textos de los comentarios por el algoritmo de Fuerza Bruta.		
Columna	Tipo	Descripción
id (PK)	int	Identificador del registro de la palabra.
opcion	int	Su valor varía en 1 y 0, y sirve para identificar si el algoritmo se encuentra activado o desactivado.
palabra	varchar(20)	La palabra a ser utilizada como patrón.

<b>TipoCocina</b>		
Tabla que almacena información con respecto a los tipos de concina con los que se puede registrar un restaurante.		
Columna	Tipo	Descripción
id (PK)	int	Identificador del registro del tipo de cocina.
descripcion	varchar(20)	Descripción del tipo de cocina.

<b>Restaurante</b>		
Tabla que almacena información de los restaurantes que se registran en el sistema.		

Columna	Tipo	Descripción
id	int	Identificador del tipo de cocina.
nombre	varchar(30)	Nombre del restaurante.
telefono	varchar(20)	Número de contacto del restaurante.
tipoLocalId (FK)	int	Identificador que asocia el restaurante con un tipo de local.
tipoCocinaId (FK)	int	Identificador que asocia el restaurante con un tipo de cocina.
distritoId (FK)	int	Identificador que asocia el restaurante con distrito.
direccion	varchar(40)	Dirección del local del restaurante.
longitud	float	Valor que indica la longitud en la que se ubica el restaurante en el globo terráqueo
latitud	float	Valor que indica la latitud en la que se ubica el restaurante en el globo terráqueo.
valorComida	int	Valor que indica la calidad de la comida, el cual se obtiene con el promedio simple de todas las calificaciones.
valorServicio	int	Valor que indica la calidad del servicio de atención, el cual se obtiene con el promedio simple de todas las calificaciones.
valorAmbiente	int	Valor que indica la calidad del ambiente, el cual se obtiene con el promedio simple de todas las calificaciones.
precioPromedio	int	Valor que indica cuánto cuesta en promedio una entrada más un plato de fondo más una bebida, el cual se obtiene con el promedio simple de todas las calificaciones.
cantidadVotos	int	Cantidad de críticas que se han registrado hacia un restaurante.
usuarioId (FK)	int	Identificador que asocia el restaurante con un usuario para que este se pueda autenticar en el

		sistema
imageData	varbinary(max)	Cadena de bytes que almacena la información de la imagen para luego poder ser renderizada.
imageMimeType	varchar(20)	Texto que indica la extensión de la imagen para luego poder ser decodificada.
fechaApertura	date	Fecha en la que inauguró el restaurante.

### MedioXRestaurante

Tabla que almacena las relaciones entre los restaurantes y los medios de pago..

Columna	Tipo	Descripción
id (PK)	int	Identificador de la relación
medioPagoId (FK)	int	Identificador del medio de pago con el que se relaciona el restaurante.
restauranteId (FK)	Int	Identificador del restaurante con el que el medio de pago se relaciona.

### MedioPago

Tabla que almacena la información de los medios de pago con los que se pueden registrar los restaurantes.

Columna	Tipo	Descripción
id (PK)	int	Identificador del medio de pago.
descripcion	varchar(20)	Descripción del medio de pago (efectivo, visa, entre otros.).

### ServicioXRestaurante

Tabla que almacena las relaciones entre los restaurantes y los servicio que brinda..

Columna	Tipo	Descripción
id (PK)	int	Identificador de la relación
servicioId (FK)	int	Identificador del servicio con el que se relaciona el

		restaurante.
restauranteId (FK)	Int	Identificador del restaurante con el que el servicio se relaciona.

<b>Servicio</b>		
Tabla que almacena la información de los servicios con los que se pueden registrar los restaurantes.		
Columna	Tipo	Descripción
id (PK)	int	Identificador del servicio.
descripcion	varchar(20)	Descripción del servicio (música en vivo, buffet, entre otros).

<b>TipoLocal</b>		
Tabla que almacena la información de los tipos de locales con los que se pueden registrar un restaurante.		
Columna	Tipo	Descripción
id (PK)	int	Identificador del tipo de local.
descripcion	varchar(20)	Descripción del tipo de local (bar, restaurante formal, entre otros).

<b>Plato</b>		
Tabla que almacena la información de los platos con los que cuentan los restaurantes.		
Columna	Tipo	Descripción
id (PK)	int	Identificador del plato.
nombre	varchar(20)	Nombre del plato.
descripcion	varchar(250)	Descripción del plato.
restauranteId (FK)	int	Identificador que asocia el plato con un restaurante.
imageData	varbinary(max)	Cadena de bytes que almacena la información de la imagen para luego poder ser renderizada.

imageMimeType	varchar(20)	Texto que indica la extensión de la imagen para luego poder ser decodificada.
---------------	-------------	-------------------------------------------------------------------------------

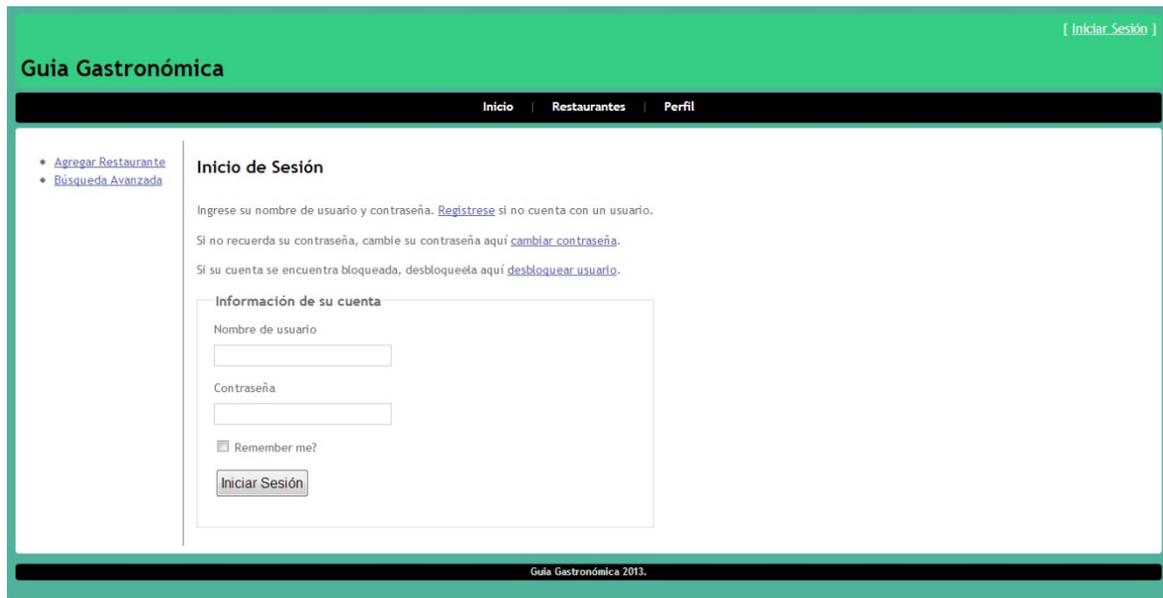
Distrito		
Tabla que almacena la información de los distritos de las distintas ciudades.		
Columna	Tipo	Descripción
id (PK)	int	Identificador del distrito.
descripcion	varchar(40)	Descripción del distrito.
ciudadId (FK)	int	Identificador que asocia un distrito con una ciudad.

Ciudad		
Tabla que almacena la información de las ciudades con las que los distritos se pueden asociar.		
Columna	Tipo	Descripción
id (PK)	int	Identificador de la ciudad.
nombre	varchar(20)	Nombre de la ciudad

## 6. Interfaces

En este punto, se muestran interfaces adicionales que no se mostraron en el documento de tesis. Las interfaces finales podrán ser visualizadas en el producto del proyecto.

## 6.1 Paquete de Seguridad



The screenshot shows the login page of the 'Guía Gastronómica' website. The page has a green header with the title 'Guía Gastronómica' and a '[ Iniciar Sesión ]' link. Below the header is a navigation bar with 'Inicio', 'Restaurantes', and 'Perfil' links. The main content area is titled 'Inicio de Sesión' and contains the following text: 'Ingrese su nombre de usuario y contraseña. [Regístrese](#) si no cuenta con un usuario. Si no recuerda su contraseña, cambie su contraseña aquí [cambiar contraseña](#). Si su cuenta se encuentra bloqueada, desbloqueeela aquí [desbloquear usuario](#).' Below this text is a form titled 'Información de su cuenta' with two input fields: 'Nombre de usuario' and 'Contraseña'. There is also a checkbox labeled 'Remember me?' and an 'Iniciar Sesión' button. At the bottom of the page, there is a footer that reads 'Guía Gastronómica 2013.'

Figura 5 “Interfaz – Iniciar Sesión” (Elaboración Propia)

## 6.2 Paquete de Comensales

[ Iniciar Sesión ]

**Guía Gastronómica**

Inicio | Restaurantes | Perfil

- [Agregar Restaurante](#)
- [Búsqueda Avanzada](#)

### Nueva cuenta

Use el formulario de abajo para crear su cuenta.  
La contraseña debe tener mínimo una cantidad de 6 caracteres.

**Información de su cuenta**

Nombre de usuario

Correo electrónico

Contraseña

Confirmación de contraseña

Nombres

Apellidos

Guía Gastronómica 2013.

Figura 6 “Interfaz – Registrar Comensal” (Elaboración Propia)

Bienvenido xtian! [ Log Off ]

**Guía Gastronómica**

Inicio | Restaurantes | Perfil

- [Agregar Restaurante](#)
- [Búsqueda Avanzada](#)

### Restaurantes

Nombre	Cocina	Distrito	Dirección	Comida	Servicio	Ambiente	Votos	Distancia
<a href="#">Don Katsu</a>	Japonesa	Victoria	Calle Santa Catalina 208	3	4	2	1	0.17 km
<a href="#">Doña Lore</a>	Japonesa	Victoria	Calle Santa Catalina 208	0	0	0	0	1.55 km
<a href="#">Doña Miki</a>	Japonesa	Victoria	Calle Santa Catalina 208	0	0	0	0	1.68 km

Guía Gastronómica 2013.

Figura 7 “Interfaz – Consultar Restaurantes” (Elaboración Propia)

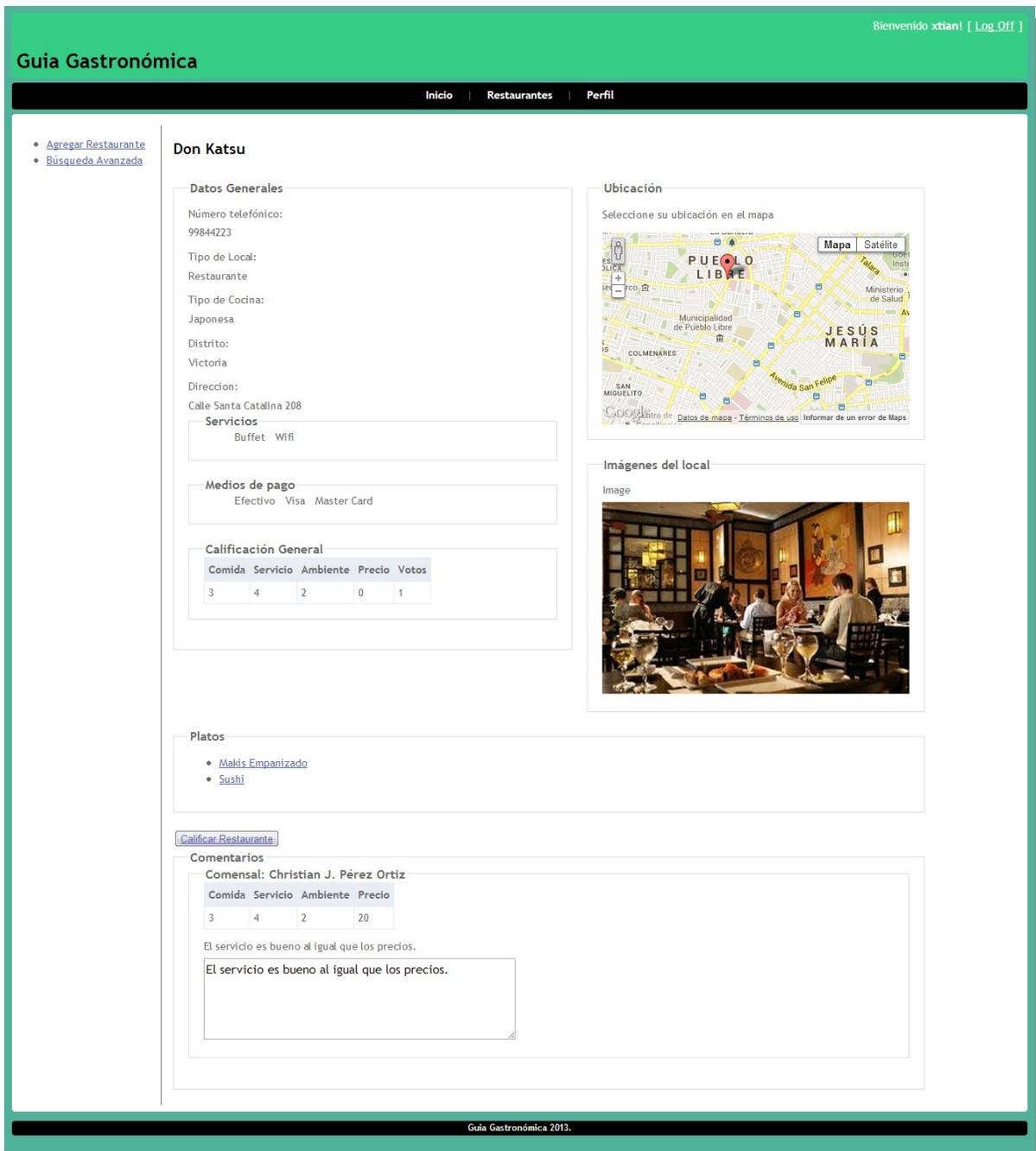


Figura 8 “Interfaz – Detalle de Restaurante” (Elaboración Propia)

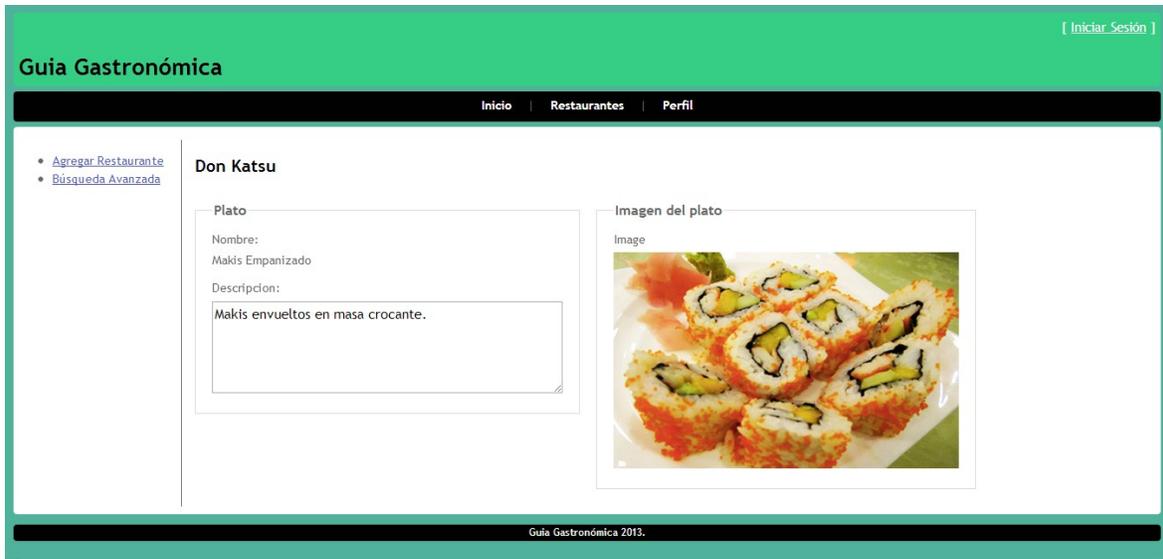


Figura 9 “Interfaz – Detalle Plato” (Elaboración Propia)

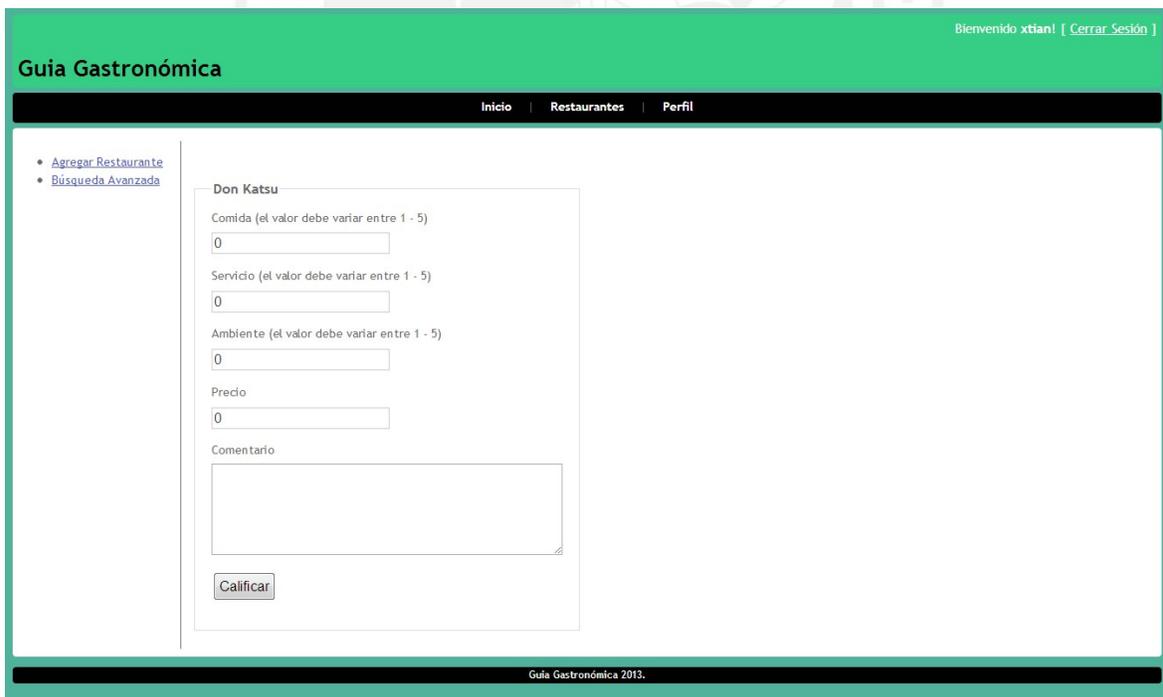


Figura 10 “Interfaz – Calificar Restaurante” (Elaboración Propia)

Bienvenido xtian! [ Cerrar Sesión ]

## Guía Gastronómica

[Inicio](#) | [Restaurantes](#) | [Perfil](#)

- [• Agregar Restaurante](#)
- [• Búsqueda Avanzada](#)

### Christian J. Pérez Ortiz

**Críticas**

Restaurante: Don Katsu

Comida	Servicio	Ambiente	Precio
3	4	2	20

Comentario:

El servicio es bueno al igual que los precios.

Estado:

Aprobado

Restaurante: Don Katsu

Comida	Servicio	Ambiente	Precio
3	3	3	50

Comentario:

esta bueno

Estado:

Aprobado

Guía Gastronómica 2013.

Figura 11 “Interfaz – Consultar Perfil” (Elaboración Propia)

### 6.3 Paquete de Restaurante

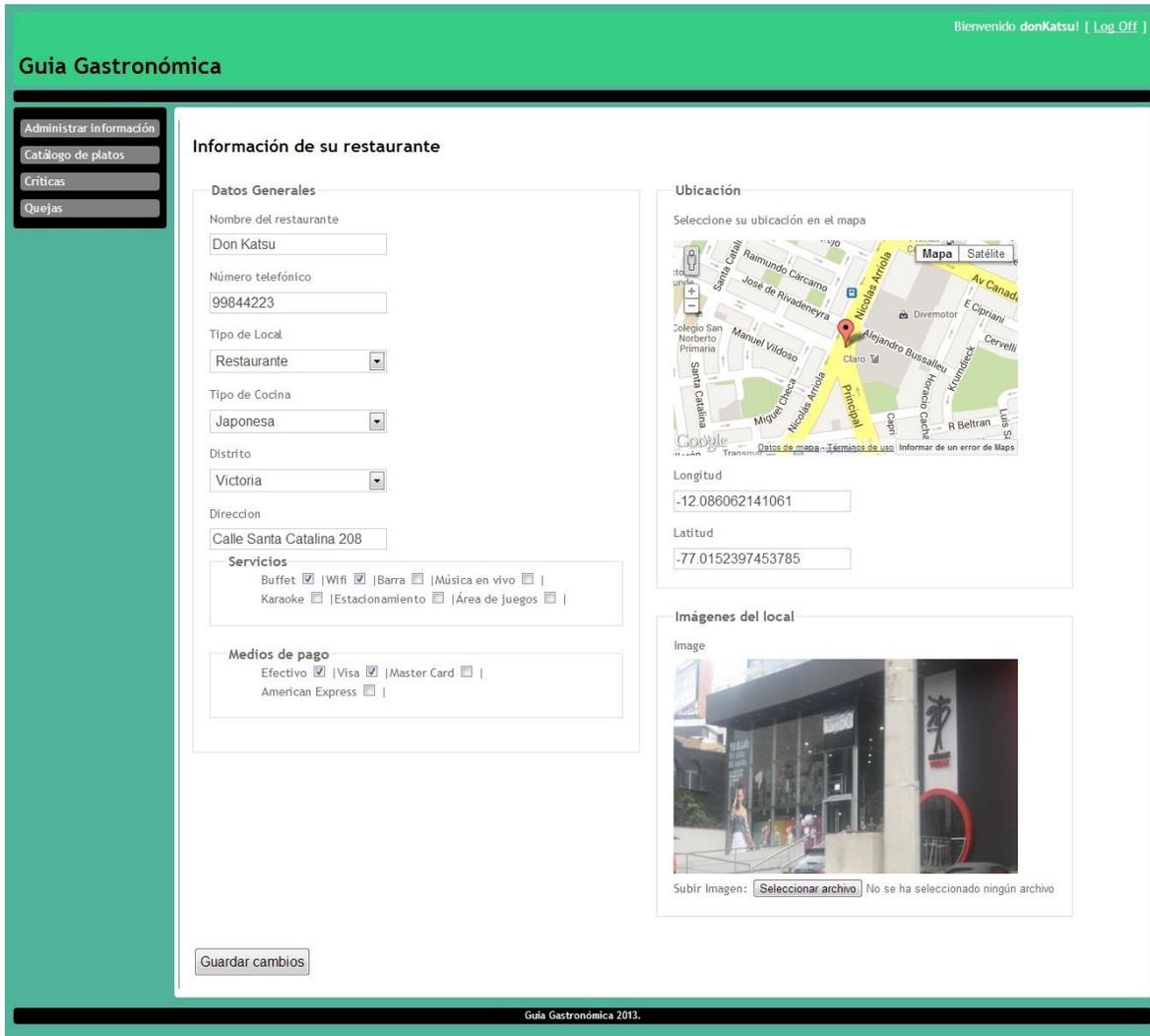


Figura 12 “Interfaz – Administrar Información del Restaurante” (Elaboración Propia)

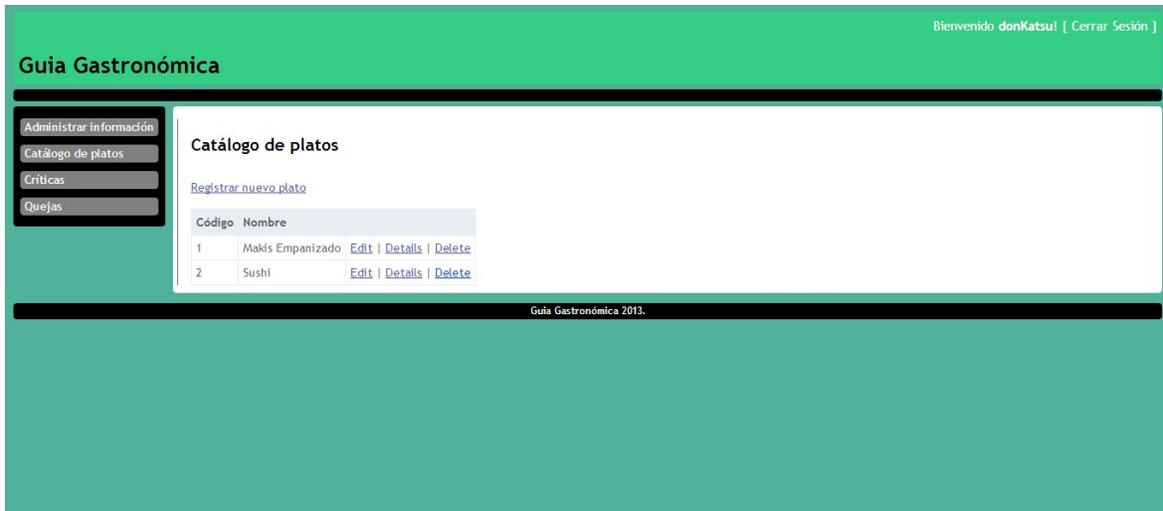


Figura 13 “Interfaz – Consultar Catálogo de Platos” (Elaboración Propia)



Figura 14 “Interfaz – Editar Plato” (Elaboración Propia)

Bienvenido donKatsu! [ Cerrar Sesión ]

## Guía Gastronómica

Administrar información  
Catálogo de platos  
Críticas  
Quejas

### Críticas

**Calificación General**

Comida	Servicio	Ambiente	Precio	Votos
3	3.5	2.5	35	2

**Críticas**

Comensal: Christian J. Pérez Ortiz

Comida	Servicio	Ambiente	Precio
3	4	2	20

Comentario

El servicio es bueno al igual que los precios.

[Reportar](#)

Comensal: Christian J. Pérez Ortiz

Comida	Servicio	Ambiente	Precio
3	3	3	50

Comentario

esta bueno

[Reportar](#)

Guía Gastronómica 2013.

Figura 15 “Interfaz – Restaurante Consulta Críticas” (Elaboración Propia)

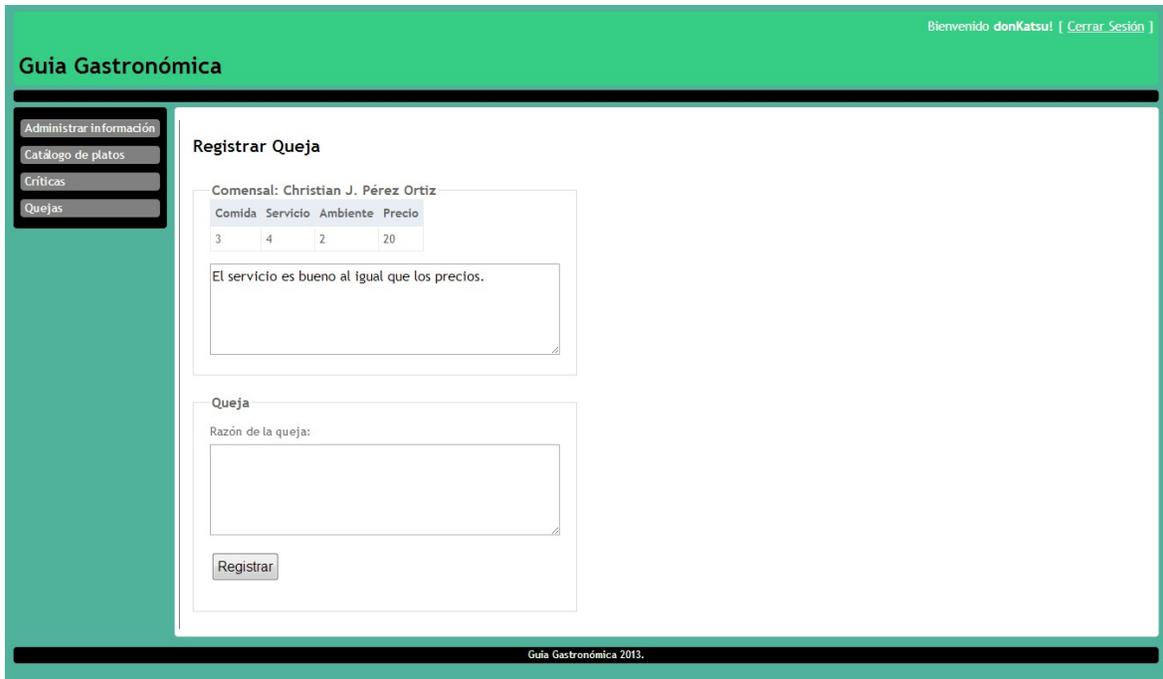


Figura 16 “Interfaz – Reportar Críticas” (Elaboración Propia)

## 6.4 Paquete de Administración

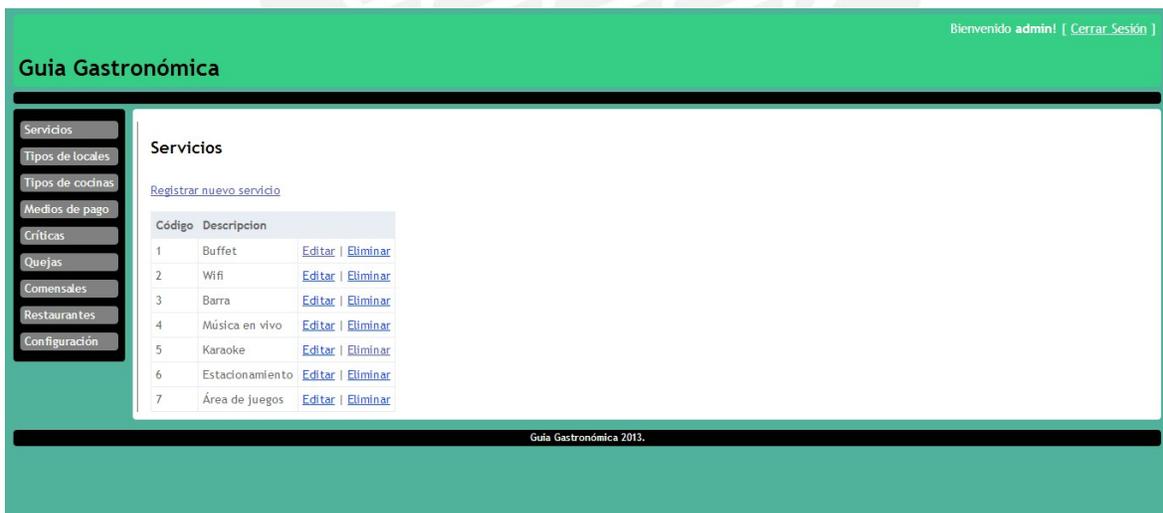


Figura 17 “Interfaz – Administrar Servicios” (Elaboración Propia)

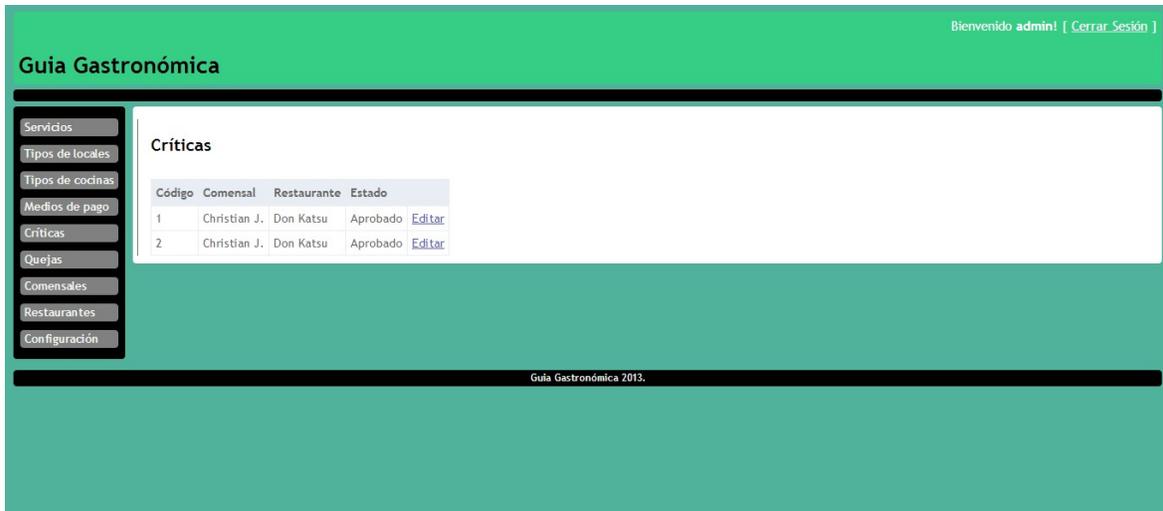


Figura 18 “Interfaz – Administrador Consulta Críticas” (Elaboración Propia)

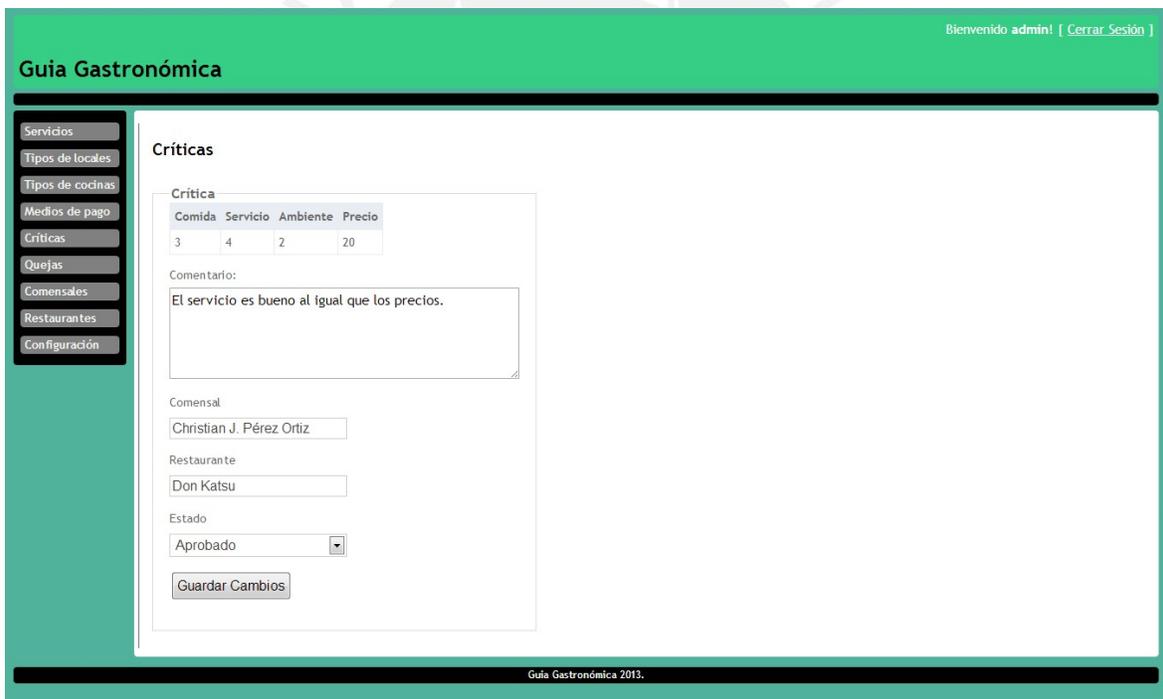


Figura 19 “Interfaz – Administrador Edita Críticas” (Elaboración Propia)

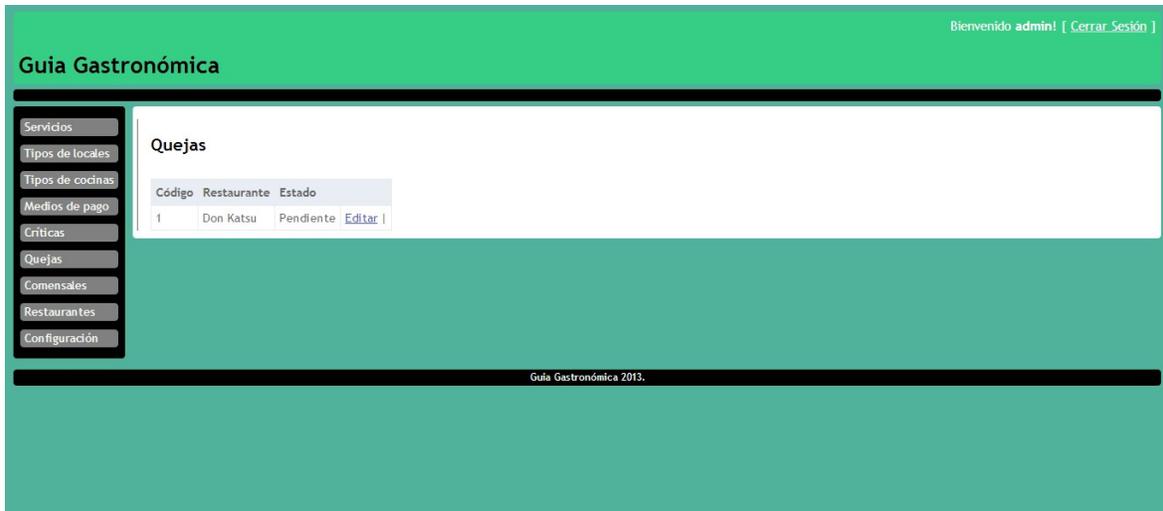


Figura 20 “Interfaz – Administrador Consulta Quejas” (Elaboración Propia)

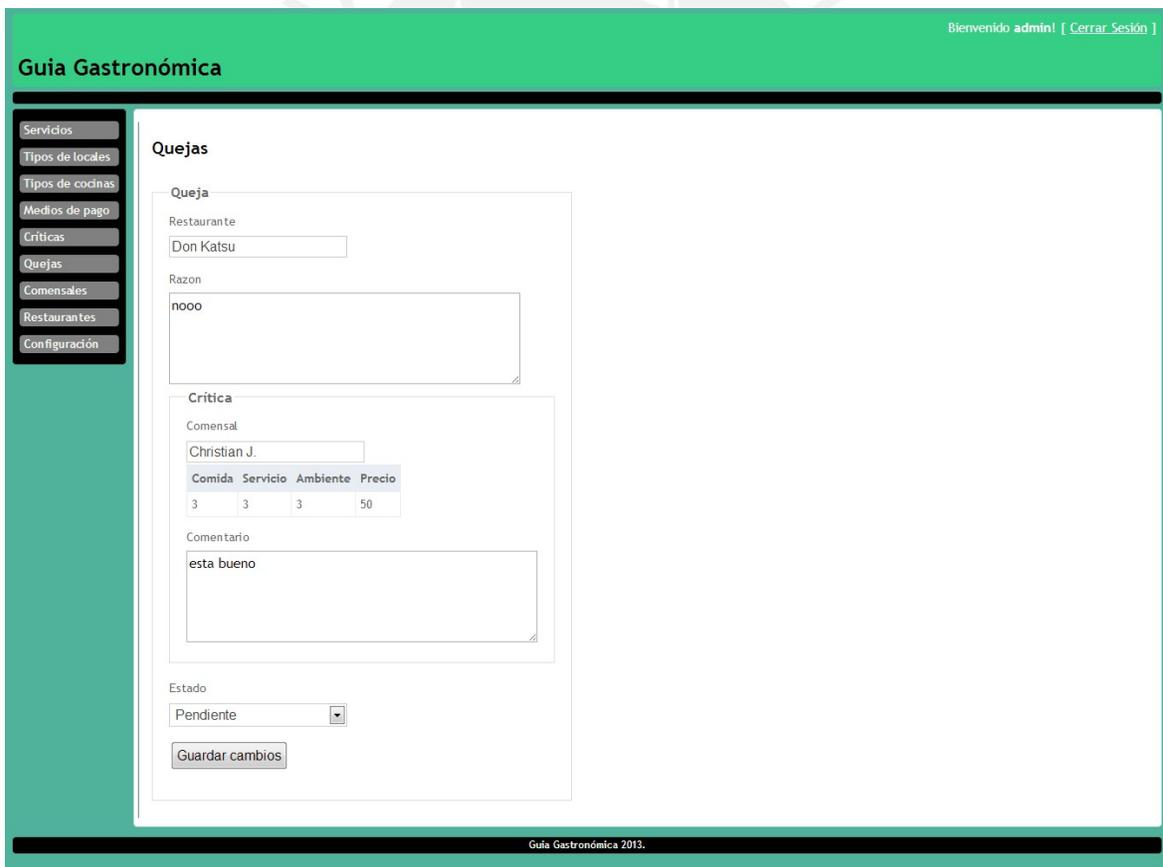


Figura 21 “Interfaz – Administrador Edita Quejas” (Elaboración Propia)



Figura 22 “Interfaz – Administrador Consulta Comensales” (Elaboración Propia)



Figura 23 “Interfaz – Administrador Edita Comensal” (Elaboración Propia)



Figura 24 “Interfaz – Administrador Consulta Restaurantes” (Elaboración Propia)

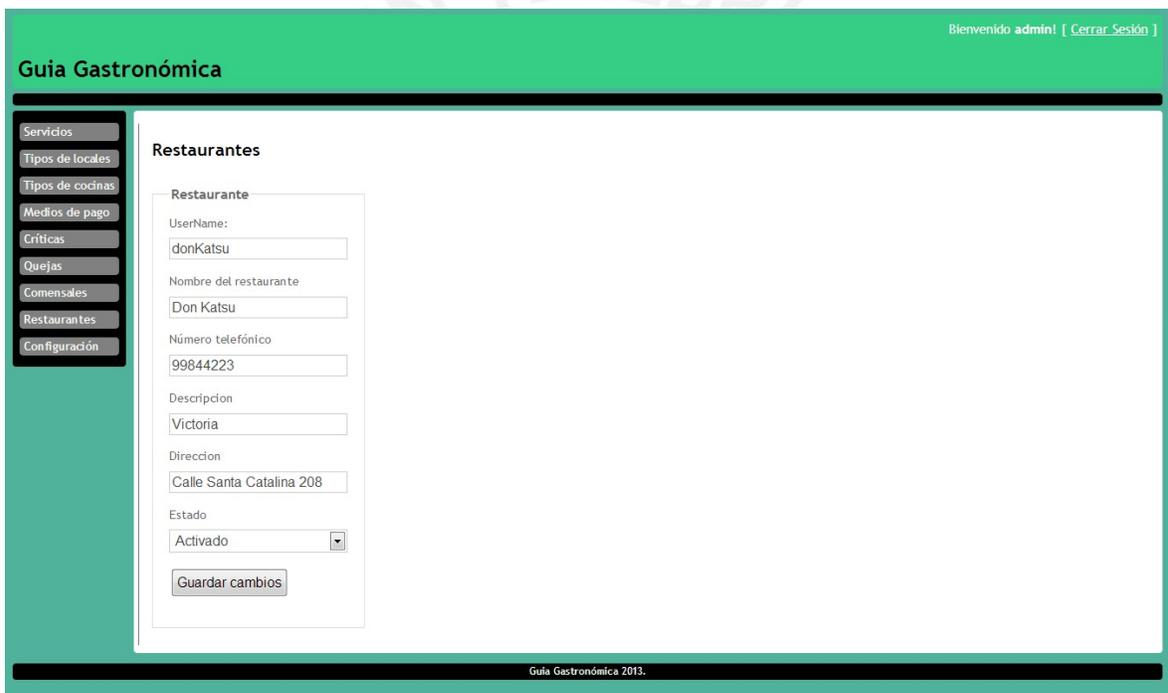


Figura 25 “Interfaz – Administrador Edita Restaurantes” (Elaboración Propia)



Figura 26 “Interfaz – Administración del Algoritmo” (Elaboración Propia)

## 7. Estándares

En este punto, se explicarán algunos de los estándares de programación que se utilizaron en el desarrollo de la implementación.

### 7.1 Llaves de bloque

En cuanto a las llaves que encierran bloques, estas deberán mostrar siempre una línea después de que se haya declara el comienzo de un bloque. Asimismo, el contenido del bloque debe empezar una línea después de donde se encuentre la llave. En la figura 27 se puede visualizar esto.

```

public ActionResult Details(int id)
{
    Restaurante restaurante = db.Restaurantes.Find(id);
    ViewBag.Critica = db.Criticas.Include(a=>a.Comensal).Where(a => a.RestauranteID == restaurante.Id)
        .Where(a=>a.Estado.Descripcion.Equals(Constants.ESTADO_APROBADO))
        .Where(a => a.Estado.Tipo.Equals(Constants.ESTADO_CRITICA)).ToList();
    return View(restaurante);
}

```

Figura 27 “Estándar - Llaves” (Elaboración Propia)

## 7.2 Comentarios

Los comentarios solo deben utilizarse para ayudar con el entendimiento del funcionamiento de ciertas partes del código, tales como las clases, las propiedades de estas, variables locales, entre otros. En la figura 28, se puede visualizar un ejemplo de cómo se ha estado utilizando los comentarios.

```
//Algoritmo de ordenamiento Shell
public static void ShellSort(List<Restaurante> listaRestaurantes)
{
    var n = listaRestaurantes.Count();
    var k = n / 2;

    while (k >= 1)
    {
        for (int i = k; i < n; i++)
        {
            var v = listaRestaurantes.ElementAt(i);
            var j = i - k;
            while (j >= 0 && listaRestaurantes.ElementAt(j).Distancia > v.Distancia)
            {
                listaRestaurantes.Insert(j + k, listaRestaurantes.ElementAt(j));
                listaRestaurantes.RemoveAt(j + k + 1);
                j = j - k;
            }
            listaRestaurantes.Insert(j + k, v);
            listaRestaurantes.RemoveAt(j + k + 1);
        }
        k = k / 2;
    }
}
```

Figura 28 “Estándar - Comentarios” (Elaboración Propia)

Un error común es dejar código comentado, esto se ha evitado en la implementación del sistema para que se pueda leer el código de manera más fluida en un futuro.

## 7.3 Espaciado

El espaciado también es un tema que ayuda con la lectura del código, por lo que se ha utilizado las siguientes reglas durante el desarrollo del sistema:

- **Utilizar** una coma entre los argumentos de una función:  
Bien: `console.In.Read(myChar, 0, 1);`  
Mal: `console.In.Read(myChar,0,1);`
- **No utilizar** espaciado justo después de un paréntesis y antes de un argumento de una función:  
Bien: `createFoo(myChar, 0, 1)`  
Mal: `createFoo( myChar, 0, 1)`
- **No utilizar** espacios entre el nombre de una función y su paréntesis:  
Bien: `createFoo()`  
Mal: `createFoo ()`
- **No utilizar** espacios dentro de los corchetes:  
Bien: `x = dataArray[index];`  
Mal: `x = dataArray[ index ];`
- **Utilizar** un espacio simple antes de una declaración de flujo de control:  
Bien: `while (x == y)`  
Mal: `while(x==y)`
- **Utilizar** un espacio simple antes y después de los operadores de comparación:  
Bien: `if (x == y)`  
Mal: `if (x==y)`

## 7.4 Nombramiento

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 4.4. 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 29 "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 4.5, 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 30 "Estándar - Parámetros" (Elaboración Propia)

## 8. Estructura del proyecto

En el proyecto se ha establecido una estructura para las capas o paquetes en los que se organizan los archivos. A continuación se empieza a mencionar cada una de estas capas.

### 8.1 Content

El paquete Content, se ha utilizado para almacenar los archivos que le brindan estilo a las interfaces web, es decir, los archivos de extensión .css.

En la figura 31, se puede visualizar que la carpeta Content posee el archivo Site.css, el cual contiene los estilos principales de las interfaces web. Asimismo, se puede apreciar una carpeta themes, la cual contiene una serie de archivos .css de jquery, los cuales se integran por defecto con el proyecto.

Así como se utilizó el archivo Site.css para el estilo de las interfaces, otro podría agregarse al proyecto y ser utilizado cambiando las referencias en los archivos html.

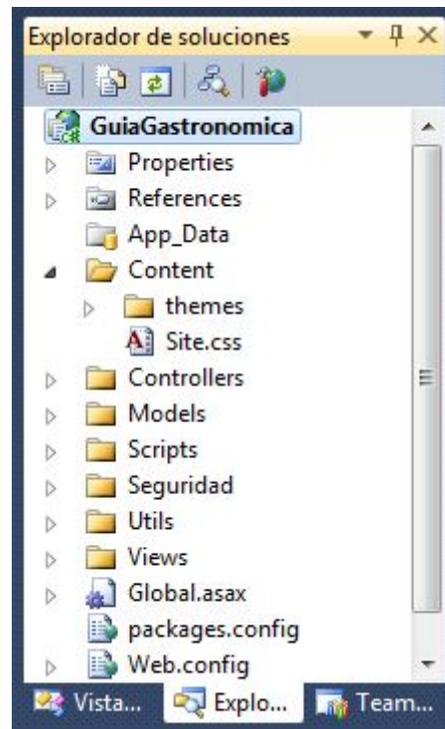


Figura 31 “Estructura del Proyecto - Content” (Elaboración Propia)

## 8.2 Controlllers

El paquete Controlllers, se ha utilizado para almacenar los archivos que permitan responder a los eventos realizados por los usuarios finales a través de las interfaces por medio de botones u otros componentes. Para esto, se utilizaron clases en C# (.cs). Parte de las convenciones para estas clases, consisten en que el nombre de la clase termine en “Controller”. Cada una de estas clases está orientada a responder eventos que estén relacionados entre ellos. Por ejemplo, la clase “RestauranteController.cs”, contiene una serie de métodos que están relacionados principalmente con los eventos que realiza un usuario tipo restaurantes. En la figura 32, se puede visualizar las clases que se utilizaron para el proyecto.

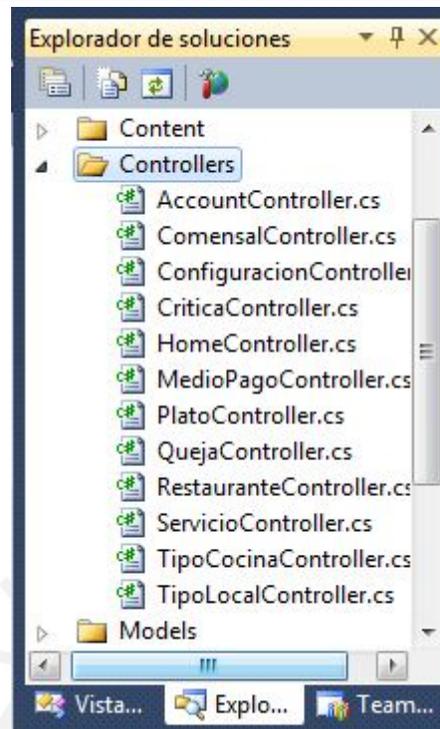


Figura 32 “Estructura del Proyecto - Controllers” (Elaboración Propia)

### 8.3 Models

El paquete Models, se ha utilizado para almacenar los archivos que permitan representar la lógica del negocio, y por ende también las entidades del negocio. Para esto, se utilizaron clases en C# (.cs). Cada una de estas clases está orientada a guardar información del negocio que esté relacionada con el nombre de la entidad. Por ejemplo, la clase “Restaurante.cs”, contiene todas las propiedades necesarias para realizar el flujo de las actividades de los usuarios restaurante. En la figura 33, se puede visualizar las clases que se utilizaron para el proyecto.

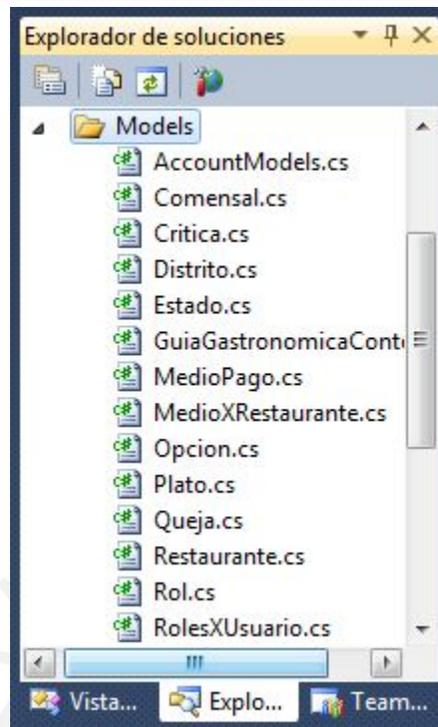


Figura 33 “Estructura del Proyecto - Models” (Elaboración Propia)

## 8.4 Scripts

El paquete Scripts, se ha utilizado para almacenar los archivos que permiten brindar mayor dinámica a las interfaces web, es decir, los archivos de extensión .js (javascript).

En la figura 34, se puede visualizar que la carpeta Script posee una serie de archivos .js de jquery. Estos hacen posible muchas de las validaciones de lado del cliente, así como la posibilidad de las llamadas asíncronas al servidor para brindar una mejor experiencia al usuario.

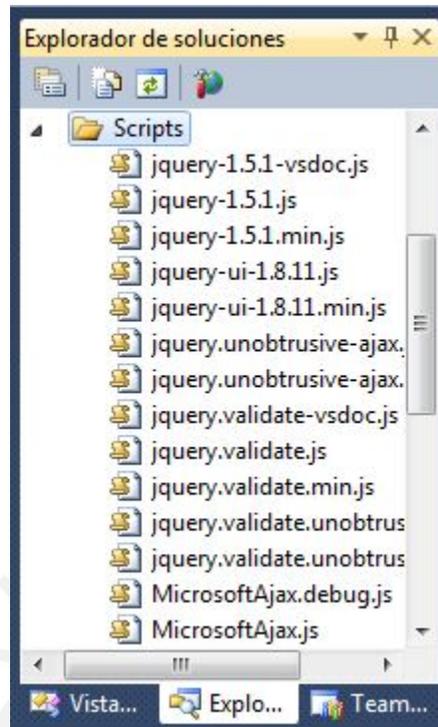


Figura 34 “Estructura del Proyecto - Scripts” (Elaboración Propia)

## 8.5 Seguridad

El paquete de Seguridad, se ha utilizado para almacenar los archivos que permiten realizar las acciones de autenticación, cambio de contraseña, entre otros que son parte de las funcionalidades de los módulos de seguridad.

Para la implementación de la seguridad, se reutilizó un mecanismo de .NET llamado Authentication Forms, el cual brinda todas las funcionalidades básicas de seguridad

En la figura 35, se puede visualizar que la carpeta Seguridad posee dos archivos en C#. Estas fueron utilizadas para sobrescribir ciertos métodos del mecanismo mencionado para poder conectarse con los datos de las tablas de usuario y roles, y así poder responder a la lógica de la guía gastronómica.

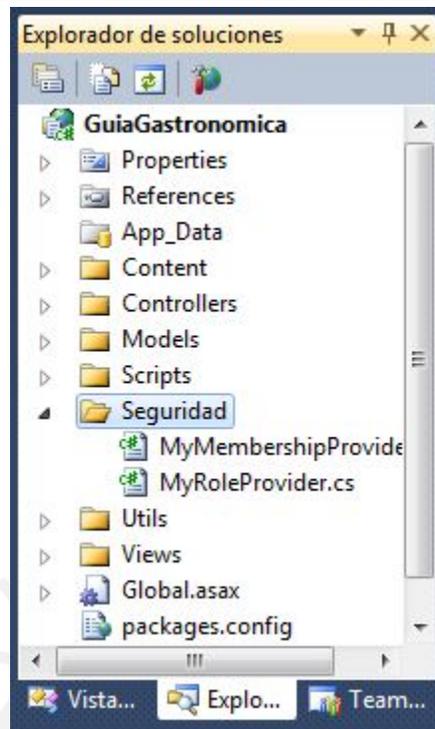


Figura 35 “Estructura del Proyecto - Seguridad” (Elaboración Propia)

## 8.6 Utils

El paquete de Utils, se ha utilizado para almacenar los archivos que contiene utilidades del sistema, principalmente los algoritmos en este caso.

En la figura 36, se puede visualizar que contiene una clase de constantes que son utilizadas por distintas clases de los controladores, modelos y vistas. Asimismo, se visualiza, un archivo llamado Math.cs, este contiene todas las propiedades relacionadas al algoritmo de ordenamiento Shell. Finalmente, el archivo Utils.cs contiene todas las propiedades relacionadas al uso del algoritmo de búsqueda de cadenas de texto Fuerza Bruta.

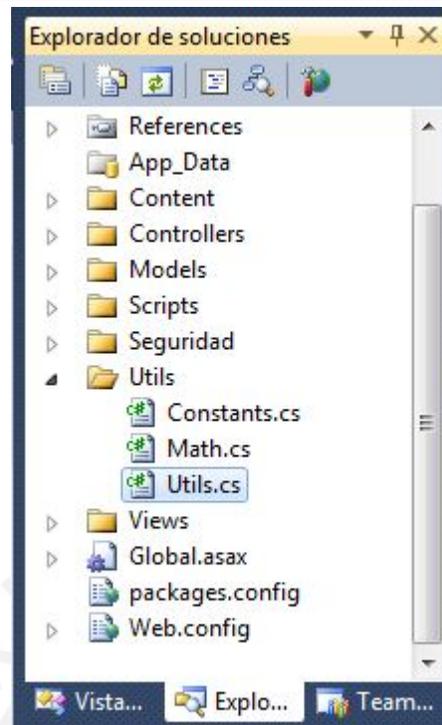


Figura 36 “Estructura del Proyecto - Utils” (Elaboración Propia)

## 8.7 Views

El paquete de Views, se ha utilizado para almacenar los archivos que generan las vistas, estos no son html directamente, sino que son archivos .cshtml que son utilizados por el motor razor para generar el código html a partir de lo que el servidor responde.

El framework asocia estos archivos con los controladores expuesto en el punto 8.2. En la figura 37, se puede visualizar que esta carpeta contiene varias carpetas con los nombres de los controladores. Cada una de estas está orientada a ser utilizada por sus controladores correspondientes, y tendrán tantas vistas como acciones y respuestas se puedan realizar sobre los controladores.

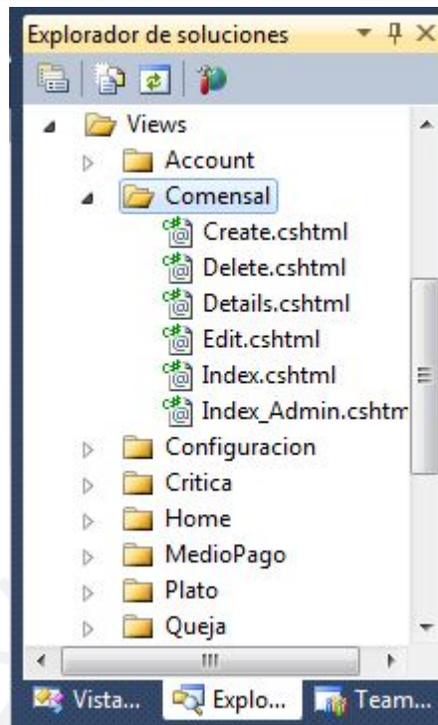


Figura 37 “Estructura del Proyecto - Views” (Elaboración Propia)

## 9. Implementación de los algoritmos

En esta sección se muestra algunos detalles acerca de la implementación de los algoritmos.

### 9.1 Algoritmo Shell

Para este algoritmo se sigue fielmente el pseudocódigo expuesto en el capítulo 4 de construcción. En la figura 38, se puede ver el código implementado del algoritmo. Algunos detalles con respecto a su implementación es que se ha utilizado colecciones de tipo listas, ya que eran los más sencillos de codificar; sin embargo, otra colección podría ser más apropiada.

Por otro lado, para el cálculo de las distancias entre los restaurantes (las cuales eran el criterio de ordenamiento para el algoritmo shell) se realizó a través de la fórmula

Haversine, la cual se muestra su implementación en la figura 39. Como se puede observar, esta tomaba como entrada las posiciones de los dos puntos (el restaurantes y el usuario), de tal forma que retornaba la distancia en kilómetros.

```

//Algoritmo de ordenamiento Shell
public static void ShellSort(List<Restaurante> listaRestaurantes)
{
    var n = listaRestaurantes.Count();
    var k = n / 2;

    while (k >= 1)
    {
        for (int i = k; i < n; i++)
        {
            var v = listaRestaurantes.ElementAt(i);
            var j = i - k;
            while (j >= 0 && listaRestaurantes.ElementAt(j).Distancia > v.Distancia)
            {
                listaRestaurantes.Insert(j + k, listaRestaurantes.ElementAt(j));
                listaRestaurantes.RemoveAt(j + k + 1);
                j = j - k;
            }
            listaRestaurantes.Insert(j + k, v);
            listaRestaurantes.RemoveAt(j + k + 1);
        }
        k = k / 2;
    }
}

```

Figura 38 “Algoritmo - Shell” (Elaboración Propia)

```

//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 39 “Fórmula Distancia” (Elaboración Propia)

## 9.2 Algoritmo Fuerza Bruta

Para este algoritmo se sigue fielmente el pseudocódigo expuesto en el capítulo 4 de construcción. En la figura 40, se puede ver el código implementado del algoritmo. Algunos detalles con respecto a su implementación es que se ha utilizado el método “ToUpperInvariant” en las letras para que al realizar las comparaciones sea irrelevante si se encuentran en mayúscula o minúscula. Por otro lado, para evitar excepciones por índice en los arreglos char[] o string, se utilizó un try catch para indicar que en caso se genere una, se considere que el patrón no ha sido encontrado aún, y así el algoritmo no se detenga.

```

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]))
                {
                    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)
                        {
                            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 40 “Algoritmo – Fuerza Bruta” (Elaboración Propia)

## 10. Casos de Prueba

Los casos de prueba, que se exponen a continuación, se han obtenido a partir de los casos de uso mostrados en el punto 2.

### 10.1 Paquete de Seguridad

Prueba de Identificación de Rol		
Datos	<b>Id</b>	TC-SEG-001
	<b>Descripción</b>	Esta prueba se centra en comprobar que la identificación del rol del usuario esté funcionando correctamente
	<b>Referencia Caso de Uso</b>	CU-SEG-001
	<b>Paquete</b>	Seguridad
	<b>Actores</b>	Consumidor, Restaurante, Administrador.
Prueba Flujo 1: “Acceso con comensal”		
<b>Acción del actor</b>		<b>Resultado Esperado</b>
1. El actor ingresa el nombre y contraseña de un usuario tipo comensal.		2. El sistema activa la sesión del usuario y mantiene la interfaz igual que la pública. Además, permite acceso a la opción de criticar y ver perfil.
Prueba Flujo 2: “Acceso con restaurante”		
1. El actor ingresa el nombre y contraseña de un usuario tipo restaurante.		2. El sistema activa la sesión del usuario, y muestra en la interfaz un menú de opciones adicional con las opciones: Administrar su información, Catálogo de platos, Críticas y Quejas.
Prueba Flujo 2: “Acceso con administrador”		
1. El actor ingresa el nombre y contraseña de un usuario tipo administrador.		2. El sistema activa la sesión del usuario, y muestra en la interfaz un menú de opciones adicional con las

	opciones: Servicios, Tipo de locales, Tipos de cocina, Medios de pago, Críticas, Quejas, Comensales, Restaurantes y Configuración.
--	------------------------------------------------------------------------------------------------------------------------------------

**Prueba de Inicio de Sesión**

Datos	<b>Id</b>	TC-SEG-002
	<b>Descripción</b>	Esta prueba se centra en comprobar que la opción de inicio de sesión esté funcionando correctamente.
	<b>Referencia Caso de Uso</b>	CU-SEG-002
	<b>Paquete</b>	Seguridad
	<b>Actores</b>	Consumidor, Restaurante, Administrador.

**Prueba Flujo 1: “Inicio de sesión”**

Acción del actor	Resultado Esperado
3. El actor debe ingresar a la opción de Iniciar Sesión.	4. El sistema debe mostrar el formulario con los campos de nombre y contraseña. Asimismo, debe mostrar enlaces para registrarse, cambiar de contraseña y desbloquear una cuenta.

**Prueba de Cambio de Contraseña**

Datos	<b>Id</b>	TC-SEG-003
	<b>Descripción</b>	Esta prueba se centra en comprobar que la opción de cambio de contraseña esté funcionando correctamente.
	<b>Referencia Caso de Uso</b>	CU-SEG-003
	<b>Paquete</b>	Seguridad
	<b>Actores</b>	Consumidor, Restaurante, Administrador.

Prueba Flujo 1: “Cambio de contraseña válido”	
Acción del actor	Resultado Esperado
1. En el punto 3 del caso de uso asociado, el actor ingresará los datos correspondientes y selecciona la opción Registrar.	2. El sistema muestra el mensaje <i>su contraseña ha sido modificada.</i>
Prueba Flujo 1: “Contraseña inválida”	
Acción del actor	Resultado Esperado
1. En el punto 3 del caso de uso asociado, el actor ingresará los datos correspondientes, pero con una contraseña que no tenga la longitud adecuada de caracteres, y selecciona la opción Registrar.	2. El sistema muestra el mensaje <i>su contraseña no es válida, por favor ingrese otra.</i>

Prueba de Desbloqueo de Acceso		
Datos	<b>Id</b>	TC-SEG-004
	<b>Descripción</b>	Esta prueba se centra en comprobar que la opción de desbloqueo esté funcionando correctamente.
	<b>Referencia Caso de Uso</b>	CU-SEG-004
	<b>Paquete</b>	Seguridad
	<b>Actores</b>	Consumidor, Restaurante, Administrador.
Prueba Flujo 1: “Cambio de contraseña válido”		
<b>Acción del actor</b>		<b>Resultado Esperado</b>
1. En el punto 3 del caso de uso asociado, el actor ingresará un correo electrónico válido y seleccionará la		2. El sistema enviará un email con una nueva contraseña.

opción Enviar.	
3. El usuario ingresará la nueva contraseña que se la he enviado.	4. El sistema permitirá el acceso al usuario.

## 10.2 Paquete de Administración

Prueba de la Administración de Críticas		
Datos	<b>Id</b>	TC-ADMIN-006
	<b>Descripción</b>	Esta prueba se centra en comprobar que la función de comprobar críticas esté operando correctamente
	<b>Referencia Caso de Uso</b>	CU-ADMIN-006
	<b>Paquete</b>	Administración del Sistema
	<b>Actores</b>	Administrador.
Prueba Flujo 1: “Aceptar Crítica”		
Acción del actor		Resultado Esperado
1. El actor selecciona la opción Críticas.		2. El sistema listará las críticas correctamente tal cual se indica en el punto 2 del CU-ADMIN-006.
3. El actor selecciona la opción Editar de cualquier crítica.		4. El sistema mostrará el detalle de la crítica de la misma forma en que se indica en el paso 4 del CU-ADMIN-006.
5. El actor asignará el estado Activado y seleccionará la opción Guardar cambios		6. El sistema mostrará correctamente el mensaje indicado en el paso 6 del CU-ADMIN-006

Prueba de Configuración de Gestión de Críticas Automatizada		
D	<b>Id</b>	TC-ADMIN-010

<b>Descripción</b>	Esta prueba se centra en el corroborar que la gestión automatizada de críticas pueda ser activada y desactivada manualmente.
<b>Referencia Caso de Uso</b>	CU-ADMIN-010
<b>Paquete</b>	Administración del Sistema
<b>Actores</b>	Administrador.

### Prueba Flujo 1: “Activar gestión automatizada”

Acción del actor	Resultado Esperado
1. El actor selecciona la opción <i>Configurar</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.

### 10.3 Paquete de Restaurante

Prueba de Administración de Información de Restaurantes		
Datos	<b>Id</b>	TC-REST-002
	<b>Descripción</b>	Esta prueba consiste en que los usuarios, de perfil restaurante, puedan administrar la información que harán pública sobre su restaurante para así darse a conocer a los comensales.
	<b>Referencia Caso de Uso</b>	CU-REST-002
	<b>Paquete</b>	Flujo Restaurantes
	<b>Actores</b>	Restaurante
Prueba Flujo 1: “Administrar Información”		
Acción del actor		Resultado Esperado
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.

### 10.4 Paquete de Consumidores

Prueba de Consulta de Restaurantes		
Datos	<b>Id</b>	TC-COM-002
	<b>Descripción</b>	Esta prueba se centra en que tanto los usuarios públicos (o no registrados) y los de perfil consumidor puedan consultar la información necesario que ellos necesiten sobre un restaurante.
	<b>Referencia Caso de Uso</b>	CU-COM-002
	<b>Paquete</b>	Flujo Consumidores
	<b>Actores</b>	Consumidor, Usuario Público
Prueba Flujo 1: “Búsqueda Normal”		
Acción del actor	Resultado Esperado	
1. El actor debe ingresar el nombre de algún restaurante en el <i>Buscador</i> .	2. El sistema muestra correctamente un listado de restaurantes con los datos que se mencionan en el punto 2 del caso de uso en referencia.	
Prueba Flujo 2: “Búsqueda Avanzada”		
1. El actor selecciona la opción <i>Búsqueda Avanzada</i> .	2. El sistema muestra los siete criterios como se menciona en el punto 2 del caso de uso en referencia. Finalmente, se muestran dos opciones: <i>Buscar</i> y <i>Limpiar</i> .	
3. El actor debe seleccionar todos los criterios que desee según sus necesidades y seleccionar la opción <i>Buscar</i> .	4. El sistema mostrará el mismo resultado del punto 2 en el flujo principal.	

Prueba de Criticar Restaurantes		
Datos	<b>Id</b>	TC-COM-005
	<b>Descripción</b>	Esta prueba se centra en que solo los usuarios de perfil consumidor pueden criticar un restaurante en registrar dicha crítica.
	<b>Referencia Caso de Uso</b>	CU-COM-005
	<b>Paquete</b>	Flujo Consumidores
	<b>Actores</b>	Consumidor
Prueba Flujo 1: “Calificar Restaurante”		
Acción del actor		Resultado Esperado
1. El actor consulta el detalle de un restaurante siguiendo el flujo del caso de uso CU-COM-003.		2. El sistema mostrará el detalle de todo el restaurante.
3. El actor selecciona la opción <i>Calificar Restaurante</i> .		4. El sistema mostrará el <i>nombre del restaurante</i> . Asimismo, mostrará los criterios de evaluación tal como se menciona en el caso de uso asociado.
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 correspondiente del caso de uso asociado y registrara la crítica para futura revisión en el sistema.

## 11. Ejecución de Pruebas

Para la ejecución se ha organizado los casos de prueba en las distintas fases: casos de uso, unitaria e integración.

### 11.1 Fase Casos de Uso

Ejecución de la Administración de Críticas		
Datos	<b>Id</b>	TCR-ADMIN-006
	<b>Referencia Caso de Prueba</b>	TC-ADMIN-006
	<b>Fase de Prueba</b>	Casos de Uso
	<b>Resultado</b>	El caso de prueba fue exitoso, el sistema mostró el detalle de la crítica y guardó satisfactoriamente el nuevo estado de esta.
	<b>Fecha</b>	06/2013

Ejecución de Configuración de Gestión de Críticas Automatizada		
Datos	<b>Id</b>	TCR-ADMIN-010
	<b>Referencia Caso de Prueba</b>	TC-ADMIN-010
	<b>Fase de Prueba</b>	Casos de Uso
	<b>Resultado</b>	El caso de prueba fue exitoso, el sistema activó el mecanismo para automatizar la administración de críticas y mostró cuántas críticas fueron aceptadas y rechazadas.
	<b>Fecha</b>	06/2013

Ejecución de Consulta de Restaurantes		
Datos	<b>Id</b>	TCR-COM-002
	<b>Referencia Caso de Prueba</b>	TC-COM-002
	<b>Fase de Prueba</b>	Casos de Uso
	<b>Resultado</b>	El caso de prueba fue exitoso, el sistema mostró el listado de restaurantes correctamente.
	<b>Fecha</b>	06/2013

Ejecución de Criticar Restaurante		
Datos	<b>Id</b>	TCR-COM-005
	<b>Referencia Caso de Prueba</b>	TC-COM-005
	<b>Fase de Prueba</b>	Casos de Uso
	<b>Resultado</b>	El caso de prueba fue exitoso, el sistema registró la crítica para futura revisión y mostró un mensaje de éxito al usuario.
	<b>Fecha</b>	06/2013

## 11.2 Fase Unitaria

Ejecución Paquete de Administración 1		
Datos	<b>Id</b>	TR-ADMIN-001
	<b>Referencia Caso de Prueba</b>	TC-ADMIN-006, TC-ADMIN-010
	<b>Fase de Prueba</b>	Unitaria
	<b>Resultado</b>	El caso de prueba fue exitoso, luego de mostrar correctamente el resultado del caso de prueba 006, mostró correctamente el resultado del caso de prueba 010 del paquete de administración.
	<b>Fecha</b>	06/2013

Ejecución Paquete de Consumidores 1		
Datos	<b>Id</b>	TR-COM-001
	<b>Referencia Caso de Prueba</b>	TC-COM-002, TC-COM-005
	<b>Fase de Prueba</b>	Unitaria
	<b>Resultado</b>	El caso de prueba fue exitoso, luego de mostrar correctamente el resultado del caso de prueba 002, mostró correctamente el resultado del caso de prueba 005 del paquete de consumidores.
	<b>Fecha</b>	06/2013

### 11.3 Fase Integración

Ejecución de Integración 1		
Datos	<b>Id</b>	TR-001
	<b>Referencia Caso de Prueba</b>	TC-SEG-001, TC-SEG-002, TC-COM-002, TC-COM-005.
	<b>Fase de Prueba</b>	Integración
	<b>Resultado</b>	El caso de prueba mostró inconsistencias al momento de empezar a criticar un restaurante, ya que no identificó el rol del usuario, por lo que el usuario público tenía acceso a una funcionalidad que no debía.
	<b>Fecha</b>	06/2013

Ejecución de Integración 2		
Datos	<b>Id</b>	TR-002
	<b>Referencia Caso de Prueba</b>	TC-SEG-001, TC-SEG-002, TC-REST-002, TC-ADMIN-006.
	<b>Fase de Prueba</b>	Integración
	<b>Resultado</b>	El caso de prueba fue exitoso, ya que al terminar el flujo del caso de uso del restaurante, se empezó el caso de uso del administrador y el sistema automáticamente solicitó credenciales a través de los casos de uso 001 y 002 de seguridad.
	<b>Fecha</b>	06/2013