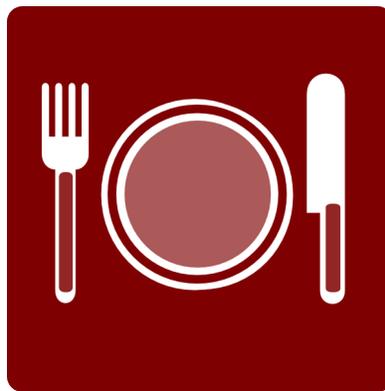


Trabajo Fin de Grado

FastService



Autor: Jon Ander Fontán Valdelvira

Grado: Ingeniería Informática de Gestión y Sistemas de Información

7 de septiembre de 2014

Resumen

El proyecto FastService, se trata de un sistema de gestión de pedidos para los restaurantes. La idea es que un usuario se pueda sentar en la mesa del restaurante y sin tener la necesidad de llamar al camarero pueda realizar los pedidos a través de su dispositivo Android.

FastService consta de tres aplicaciones:

- Aplicación Android
 - Se trata de una aplicación capaz de realizar pedidos, tanto de los artículos como de los menús que se encuentren disponibles en el restaurante. Esta aplicación alertará al usuario cuando el camarero realice alguna acción sobre su pedido, como puede ser el aviso de que su pedido ha sido atendido.
- Aplicación Web
 - Se trata de una aplicación web, la cual permite al camarero gestionar todo lo referente a su restaurante, así como atender los diferentes pedidos realizados a través de la aplicación Android.
- Aplicación Servidora
 - Aplicación que interconecta las aplicaciones de usuario con la base de datos MySQL. Esta aplicación esta basada en la tecnología Node.js que permite el uso de JavaScript en servidor.

Todos los datos de la aplicación se almacenan en una base de datos MySQL que proporciona la capacidad necesaria para esta aplicación.

Índice general

Introducción	10
Objetivos del proyecto	11
Objetivos	11
Propósito	11
Ámbito	11
Contexto de Negocio	11
Definiciones, acrónimos, y abreviaturas	11
Alcance	12
Estructura de Descomposición del Trabajo (EDT)	13
Desglose de Tareas	13
Planificación Temporal	19
Herramientas	20
Gestión de Riesgos	22
Evaluación Económica	24
Antecedentes	25
Situación Actual	25
Estudio de Diferentes Alternativas Existentes	25
Captura de Requisitos	26
Modelo de Casos de Uso	26
Jerarquía de Actores	27
Prototipos de Interfaces de Usuario	27
Modelo de Dominio	35
Descripción de la Arquitectura	35

Análisis y Diseño	36
Transformación de Modelo de Dominio a Base de Datos	36
Diagrama de Clases	37
Arquitectura del Sistema	41
Diagrama de Secuencias	41
Desarrollo	42
Verificación y Evaluación	43
Pruebas del Software	43
Pruebas de Aceptación	43
Documentación de Pruebas	43
Conclusiones y Trabajo Futuro	46
Bibliografía	47
ANEXO I.- Casos de Uso Extendidos	48
Caso de Uso Extendido Android: Iniciar Sesión	48
Caso de Uso Extendido Android: Registrarse	50
Caso de Uso Extendido Android: Realizar Pedido	52
Caso de Uso Extendido Android: Enviar Pedido	57
Caso de Uso Extendido Android: Solicitar Cobro	59
Caso de Uso Extendido Web: Iniciar Sesión	61
Caso de Uso Extendido Web: Registrarse	63
Caso de Uso Extendido Web: Gestionar Restaurante	65
Caso de Uso Extendido Web: Gestionar Carta	69
Caso de Uso Extendido Web: Gestionar Menú	72
Caso de Uso Extendido Web: Atender Pedido	75
Caso de Uso Extendido Web: Realizar Cobro	77
ANEXO II.- Diagramas de Secuencia	78
Diagrama de Secuencia Android: Registrarse	78
Diagrama de Secuencia Android: Iniciar Sesión	79
Diagrama de Secuencia Android: Realizar Pedido	80
Diagrama de Secuencia Android: Enviar Pedido	86
Diagrama de Secuencia Android: Solicitar Cobro	87
Diagrama de Secuencia Web: Registrarse	88

Diagrama de Secuencia Web: Iniciar Sesión	89
Diagrama de Secuencia Web: Gestionar Restaurante	90
Diagrama de Secuencia Web: Gestionar Carta	92
Diagrama de Secuencia Web: Gestionar Menús	94
Diagrama de Secuencia Web: Atender Pedido	96
Diagrama de Secuencia Web: Gestionar Cobro	97
ANEXO III.- Manual de usuario	98
Diagrama de Secuencia Web: Instalación	98
Diagrama de Secuencia Web: Manual	99
ANEXO IV.- Manual de administrador	111
Diagrama de Secuencia Web: Instalación	111
Diagrama de Secuencia Web: Manual	111

Índice de figuras

1.	Diagrama de Estructura de Descomposición del Trabajo	13
2.	Diagrama de Gantt	19
3.	Logo Android Studio	20
4.	Logo Sublime Text Editor	20
5.	Logo Lyx	20
6.	Logo PHPMyAdmin	21
7.	Logo Visio	21
8.	Logo Cacao	21
9.	Logo Dropbox	21
10.	Casos de Uso	26
11.	Jerarquía de Actores	27
12.	Boceto Android Inicio Sesión	27
13.	Boceto Android Registro	28
14.	Boceto Android Pedido	28
15.	Boceto Android Menú	29
16.	Boceto Android Pedido Popup	29
17.	Boceto Android Selección Artículos	30
18.	Boceto Android Selección Artículos (Horizontal)	30
19.	Boceto Android Selección Artículos (Horizontal)	30
20.	Boceto Android Historial	31
21.	Boceto Android Historial (Horizontal)	31
22.	Boceto Android Notificación	31
23.	Boceto Web Iniciar Sesión	32
24.	Boceto Web Registro	32
25.	Boceto Web Restaurante	32
26.	Boceto Web Restaurante Popup	33
27.	Boceto Web Carta	33

28.	Boceto Web Menús	34
29.	Boceto Web Pedidos	34
30.	Modelo de Dominio	35
31.	Esquema de la Arquitectura del Sistema	35
32.	Diagrama Relacional	36
33.	Diagrama de Clases Parte 1	37
34.	Diagrama de Clases Parte 2	38
35.	Diagrama de Clases Parte 3	39
36.	Diagrama de Clases Relaciones	40
37.	Arquitectura del Sistema	41
38.	CDU Extendido Android: Iniciar Sesión	48
39.	Inicio Sesión	49
40.	CDU Extendido Android: Registrarse	50
41.	Inicio Sesión	51
42.	Registro	51
43.	CDU Extendido Android: Realizar Pedido	52
44.	Pedido Vacío	54
45.	Pedido Lleno	54
46.	Selección Artículo	55
47.	Lista Artículos	55
48.	Detalle Artículo	55
49.	Selección Artículo Menú	56
50.	Lista Artículos Tipo Plato	56
51.	CDU Extendido Android: Enviar Pedido	57
52.	Pedido	58
53.	CDU Extendido Android: Solicitar Cobro	59
54.	Pedido	60
55.	CDU Extendido Web: Iniciar Sesión	61
56.	Inicio Sesión	62
57.	CDU Extendido Web: Registrarse	63
58.	Inicio Sesión	64
59.	Registro	64
60.	CDU Extendido Web: Gestionar Restaurante	65
61.	Restaurante 1	67
62.	Restaurante 2	67

63.	Restaurante 3	68
64.	CDU Extendido Web: Gestionar Carta	69
65.	Carta 1	70
66.	Carta 2	71
67.	CDU Extendido Web: Gestionar Menú	72
68.	Menú 1	73
69.	Menú 2	74
70.	CDU Extendido Web: Atender Pedido	75
71.	Pedidos 1	76
72.	Pedidos 2	76
73.	CDU Extendido Web: Realizar Cobro	77
74.	Pedidos 1	77
75.	Diagrama de Secuencia Android: Registrarse	78
76.	Diagrama de Secuencia Android: Iniciar Sesión	79
77.	Diagrama de Secuencia Android: Realizar Pedido 1/6	80
78.	Diagrama de Secuencia Android: Realizar Pedido 2/6	81
79.	Diagrama de Secuencia Android: Realizar Pedido 3/6	82
80.	Diagrama de Secuencia Android: Realizar Pedido 4/6	83
81.	Diagrama de Secuencia Android: Realizar Pedido 5/6	84
82.	Diagrama de Secuencia Android: Realizar Pedido 6/6	85
83.	Diagrama de Secuencia Android: Enviar Pedido	86
84.	Diagrama de Secuencia Android: Solicitar Cobro	87
85.	Diagrama de Secuencia Web: Registrarse	88
86.	Diagrama de Secuencia Web: Iniciar Sesión	89
87.	Diagrama de Secuencia Web: Gestionar Restaurante 1/2	90
88.	Diagrama de Secuencia Web: Gestionar Restaurante 2/2	91
89.	Diagrama de Secuencia Web: Gestionar Carta 1/2	92
90.	Diagrama de Secuencia Web: Gestionar Carta 2/2	93
91.	Diagrama de Secuencia Web: Gestionar Menús 1/2	94
92.	Diagrama de Secuencia Web: Gestionar Menús 2/2	95
93.	Diagrama de Secuencia Web: Atender Pedido	96
94.	Diagrama de Secuencia Web: Gestionar Cobro	97
95.	Manual Usuario Android: Instalación 1	98
96.	Manual Usuario Android: Instalación 2	98
97.	Manual Usuario Android: Instalación 3	99

98.	Manual Usuario Android: Pantalla Principal	99
99.	Manual Usuario Android: Registro	100
100.	Manual Usuario Android: Instrucciones	100
101.	Manual Usuario Android: Escanear QR	101
102.	Manual Usuario Android: Pedido sin Artículos	101
103.	Manual Usuario Android: Mensaje al Realizar Pedido	101
104.	Manual Usuario Android: Lista Carta	102
105.	Manual Usuario Android: Lista Carta Desplegada	102
106.	Manual Usuario Android: Detalles Artículo	103
107.	Manual Usuario Android: Mensaje de Elemento Añadido al Pedido	103
108.	Manual Usuario Android: Pedido con Artículos	104
109.	Manual Usuario Android: Menú-Primer Plato	104
110.	Manual Usuario Android: Menú-Segundo Plato	104
111.	Manual Usuario Android: Menú-Postre	105
112.	Manual Usuario Android: Menú-Bebida	105
113.	Manual Usuario Android: Pedido de Menú	105
114.	Manual Usuario Android: Pedido	106
115.	Manual Usuario Android: Pedido Atendido	106
116.	Manual Usuario Android: Notificación de Pedido	106
117.	Manual Usuario Android: Artículos nuevos y Pendientes	107
118.	Manual Usuario Android: Pedido Atendido Boton Pagar Activo	107
119.	Manual Usuario Android: Notificación Después de Pulsar Pagar	108
120.	Manual Usuario Android: Menú Desplegable	108
121.	Manual Usuario Android: Menú Desplegable en Pedido	109
122.	Manual Usuario Android: Menú Desplegable en Historial	109
123.	Manual Usuario Android: Historial	110
124.	Manual Usuario Android: Detalle Pedido en Historial	110
125.	Manual Usuario Android: Ayuda	110
126.	Manual Usuario Web: Barra de dirección	111
127.	Manual Usuario Web: Pantalla Principal	111
128.	Manual Usuario Web: Editar Datos del Restaurante	112
129.	Manual Usuario Web: Pestaña Mi Restaurante	112
130.	Manual Usuario Web: Pestaña Mi Carta	112
131.	Manual Usuario Web: Añadir/Editar Plato	113
132.	Manual Usuario Web: Pestaña Mis Menús	113

133. Manual Usuario Web: Añadir Menú	113
134. Manual Usuario Web: Pestaña Atender Pedidos	113
135. Manual Usuario Web: Detalles Pedido	114
136. Manual Usuario Web: Notificación a Usuario	114
137. Manual Usuario Web: Pestaña Cobrar Pedidos	114

Índice de cuadros

1.	Desglose de Tareas: 0.1-.Documentación	13
2.	Desglose de Tareas: 1.1-. Investigación Node.js	14
3.	Desglose de Tareas: 1.2-.Investigación Android	14
4.	Desglose de Tareas: 2.1-.Diseño de Base de Datos	14
5.	Desglose de Tareas: 2.2-.Diseño de Diagramas	15
6.	Desglose de Tareas: 3.1-.Desarrollo de la Base de Datos	15
7.	Desglose de Tareas: 3.2.1-.Aplicación Node.js-Lógica de Negocio	15
8.	Desglose de Tareas: 3.2.2-.Aplicación Node.js-Depuración	16
9.	Desglose de Tareas: 3.3.1-.Aplicación Android-Interfaces	16
10.	Desglose de Tareas: 3.3.2-.Aplicación Android-Lógica de Negocio	16
11.	Desglose de Tareas: 3.3.3-.Aplicación Android-Depuración	17
12.	Desglose de Tareas: 3.4.1-.Aplicación Web-Interfaces	17
13.	Desglose de Tareas: 3.4.2-.Aplicación Web-Lógica de Negocio	17
14.	Desglose de Tareas: 3.4.3-.Aplicación Web-Depuración	18
15.	Desglose de Tareas: 4-.Manuales	18
16.	Riesgo 1	22
17.	Riesgo 2	22
18.	Riesgo 3	23
19.	Riesgo 4	23
20.	Riesgo 5	23
21.	Horas Semanales/Tarea	24
22.	Desglose de Costes	24
23.	Documentación de Pruebas Android	44
24.	Documentación de Pruebas Web	45

Introducción

En el presente documento se puede encontrar toda la documentación que se ha ido realizando durante la elaboración del proyecto FastService. Dentro del mismo, se recogerá información de la razón por la cual se decide realizar dicha aplicación, así como las distintas fases de elaboración del proyecto y de su análisis.

Este TFG en concreto ha sido elegido a raíz de las distintas experiencias obtenidas durante visitas a restaurantes. Aprecié que en los momentos en que se tenía gran cantidad de clientes, el servicio empeoraba y a su vez las esperas, por lo que se me ocurrió que se podía implementar un sistema para mejorar estos problemas.

Dicho sistema deberá ser capaz de realizar los pedidos del cliente desde su propio dispositivo Android, de modo que el camarero los atendería en un ordenador permitiendo respetar una cola de llegada de pedidos, permitiendo que los usuarios tengas un servicio con mayor orden y más eficaz.

En este TFG se aplicarán los conocimientos adquiridos durante la carrera en los aspectos de programación y gestión de proyectos para las distintas aplicaciones a desarrollar así como para la realización del presente documento.

Objetivos del proyecto

Objetivos

El TFG FastService consiste en la implementación de un sistema de gestión de pedidos para los restaurantes.

Dicho sistema constará de dos aplicaciones, una de ellas para dispositivos móviles y otra en un entorno web, además el sistema contará con una aplicación servidora.

La aplicación para dispositivos móviles con Android, permitirá a sus usuarios identificarse en el sistema y de esta forma poder realizar pedidos.

La aplicación para entornos web, otorgará al restaurante una forma eficaz de; gestionar los artículos de su carta, configurar sus menús y recibir los pedidos de sus clientes.

La aplicación servidora será un intermediario entre las aplicaciones de usuario y la base de datos permitiendo obtener la información necesaria para el funcionamiento de ambas.

Propósito

El propósito de este proyecto es crear un sistema completo, para la gestión de los pedidos de los usuarios a los restaurantes, así como una alternativa al clásico servicio que se presta actualmente a través de un camarero.

Ámbito

La implantación de este sistema, afectaría a el colectivo de lo camareros del restaurante dado que añadiría funciones extras a su trabajo habitual, como puede ser la gestión a través del portal web de los productos que ofrece el restaurante, así como una atención constante a los distintos pedidos generados por los clientes.

Contexto de Negocio

El dominio que se intentará abarcar con este proyecto, será el colectivo de hostelería, más concretamente, el de los restaurantes.

Los usuarios de este sistema serán tanto los clientes de los restaurantes que posean un dispositivo Android, así como los empleados de dicho restaurante.

Este sistema se desarrollara como TFG de Grado en Ingeniería Informática de Gestión y Sistemas de Información con el previo acuerdo entre tutores del proyecto y se implementará desde cero.

Definiciones, Acrónimos, y Abreviaturas

TFG: Trabajo Fin de Grado.

GCM: Google Cloud Messagin.

IDE: Integrated Development Environment.

SGBD: Sistema Gestor de Base de Datos.

HTTP: Hypertext Transfer Protocol.

TPV: Terminal Punto de Venta.

HTML: HyperText Markup Language.

Alcance

La aplicación para dispositivos Android deberá cumplir las siguientes afirmaciones:

- El usuario podrá:
 - Iniciar sesión en el sistema, en caso de que no disponga de usuario deberá poder registrarse en el sistema.
 - Abrir un ticket a partir de un código QR que le identificará en el restaurante y la mesa en la que esta sentado.
 - Visualizar el pedido que esta realizando así como el estado individual de los distintos artículos o menús que haya añadido al pedido.
 - Añadir artículos del restaurante o menús previamente configurados por el menú del restaurante.
 - Modificar o eliminar los distintos artículos de su pedido siempre y cuando no los haya pedido ya.
 - Eliminar un ticket en caso exclusivo que no tenga artículos añadidos en él.
 - Cerrar la sesión en caso de que desee iniciar sesión en otro dispositivo.
 - Visualizar el historial de pedidos que ha realizado.
- El dispositivo debe permitir la recepción de notificaciones para que el usuario conozca actualizaciones sobre el estado de su pedido.

La aplicación web deberá cumplir las siguientes afirmaciones:

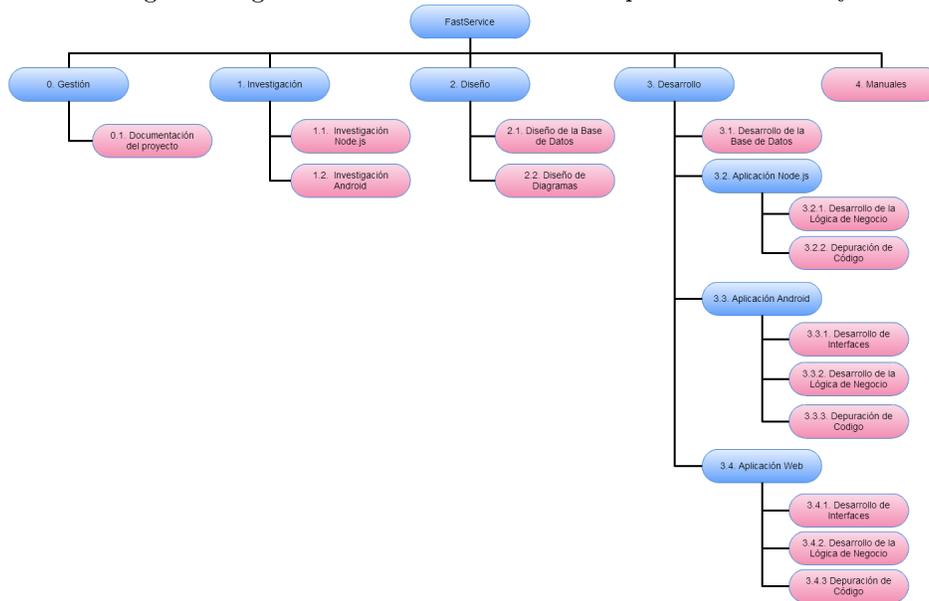
- El usuario podrá:
 - Iniciar sesión en el sistema, en caso de que no disponga de usuario deberá poder registrarse en el sistema.
 - Modificar los datos correspondientes a su restaurante y gestionar las distintas mesas que lo componen.
 - Generar el código QR de cada mesa para su uso en las mesas.
 - Visualizar los elementos que componen su carta y añadirlos, modificarlos o eliminarlos a voluntad.
 - Visualizar los menús que tiene disponibles y añadirlos, modificarlos o eliminarlos a voluntad.
 - Visualizar las distintas peticiones de pedido con sus correspondientes artículos, así como las distintas peticiones de cobro.
- La aplicación debe de mostrar ventanas emergentes en el momento de proceder a la atención de un pedido proporcionando una inserción de texto para que el usuario le envíe una notificación personalizada al cliente.

La aplicación servidora deberá cumplir las siguientes afirmaciones:

- Atender las peticiones de las aplicaciones de los usuarios y devolver los datos que se solicitan, estando estos alojados en la base de datos.
- Conectarse con el sistema de notificaciones (GCM) para poder enviarlas a los usuarios.

Estructura de Descomposición del Trabajo (EDT)

1. Figura: Diagrama de Estructura de Descomposición del Trabajo



Desglose de Tareas

1. Cuadro: Desglose de Tareas: 0.1-Documentación

<i>Paquete de trabajo:</i>	Documentación.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	20 horas/persona.
<i>Duración:</i>	20 horas.
<i>Descripción:</i>	Tarea en la cual se englobara cualquier tarea referente a la documentación. (Redacción de la memoria del proyecto y unión de los distintos anexos.)
<i>Entradas:</i>	Información de todos los campos que se van a documentar.
<i>Salidas/Entregables:</i>	Documento de la Memoria del proyecto.
<i>Recursos Necesarios:</i>	Editor de documentos de látex y una persona.
<i>Precedencias:</i>	La realización de un análisis previo a los puntos que se describen en la documentación.

2. Cuadro: Desglose de Tareas: 1.1-. Investigación Node.js

<i>Paquete de trabajo:</i>	Investigación Node.js.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	5 horas/persona.
<i>Duración:</i>	5 horas.
<i>Descripción:</i>	Investigación del funcionamiento de Node.js y de los distintos complementos necesarios para completar las necesidades del proyecto.
<i>Entradas:</i>	Ninguna.
<i>Salidas/Entregables:</i>	Decisiones sobre los complementos a usar y conocimientos sobre ellos.
<i>Recursos Necesarios:</i>	Documentación de Node.js, servidor Node.js y una persona.
<i>Precedencias:</i>	Ninguna.

3. Cuadro: Desglose de Tareas: 1.2-.Investigación Android

<i>Paquete de trabajo:</i>	Investigación Android.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	5 horas/persona.
<i>Duración:</i>	5 horas.
<i>Descripción:</i>	Investigación del funcionamiento de algunos Frameworks de Android necesarios para completar las necesidades del proyecto.
<i>Entradas:</i>	Ninguna.
<i>Salidas/Entregables:</i>	Decisiones sobre los Frameworks a usar y conocimientos sobre ellos.
<i>Recursos Necesarios:</i>	Documentación de Frameworks de Android, IDE Android Studio, un dispositivo Android y una persona.
<i>Precedencias:</i>	Ninguna.

4. Cuadro: Desglose de Tareas: 2.1-.Diseño de Base de Datos

<i>Paquete de trabajo:</i>	Diseño de Base de Datos.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	1.5 horas/persona.
<i>Duración:</i>	1.5 horas.
<i>Descripción:</i>	Realización del diseño de las diferentes tablas y relaciones necesarias para el almacenamiento de los datos en la Base de Datos.
<i>Entradas:</i>	Planificación del proyecto.
<i>Salidas/Entregables:</i>	Tablas y relaciones de la Base de Datos.
<i>Recursos Necesarios:</i>	Aplicación para realizar el diagrama de diseño de la Base de Datos y una persona.
<i>Precedencias:</i>	Diagrama de Comunicación.

5. Cuadro: Desglose de Tareas: 2.2-.Diseño de Diagramas

<i>Paquete de trabajo:</i>	Diseño de Diagramas.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	30 horas/persona.
<i>Duración:</i>	30 horas.
<i>Descripción:</i>	Realización de los distintos diagramas de Clases, Casos de Uso y Secuencia necesarios para reflejar los atributos y métodos existentes en las distintas aplicaciones del proyecto, así como el seguimiento de cada método.
<i>Entradas:</i>	Planificación del proyecto, bocetos de pantallas y tablas de la Base de Datos.
<i>Salidas/Entregables:</i>	Diagramas de Clase, Casos de Uso y Secuencia.
<i>Recursos Necesarios:</i>	Aplicación Visual Paradigm, Cacao y una persona.
<i>Precedencias:</i>	2.1 Diseño de la Base de Datos.

6. Cuadro: Desglose de Tareas: 3.1-.Desarrollo de la Base de Datos

<i>Paquete de trabajo:</i>	Desarrollo de la Base de Datos.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	2 horas/persona.
<i>Duración:</i>	2 horas.
<i>Descripción:</i>	Realización de la completa implementan de la Base de Datos utilizando el diseño de la base de datos realizado.
<i>Entradas:</i>	Diseño de la Base de Datos.
<i>Salidas/Entregables:</i>	Base de datos implementada.
<i>Recursos Necesarios:</i>	Diseño de la Base de Datos, SGBD MySQL y una persona.
<i>Precedencias:</i>	2.1 Diseño de Base de Datos.

7. Cuadro: Desglose de Tareas: 3.2.1-.Aplicación Node.js-Lógica de Negocio

<i>Paquete de trabajo:</i>	Aplicación Node.js-Lógica de Negocio.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	15 horas/persona.
<i>Duración:</i>	15 horas.
<i>Descripción:</i>	Realización de la completa implementación de toda la lógica de negocio de la aplicación servidora basada en Node.js.
<i>Entradas:</i>	Diseño de la Base de Datos.
<i>Salidas/Entregables:</i>	Aplicación servidora Node.js.
<i>Recursos Necesarios:</i>	Editor HTML/Javascript, Conexión a la base de Datos MySQL y una persona.
<i>Precedencias:</i>	1.1-. Investigación Node.js, 3.1 Desarrollo de la Base de Datos.

8. Cuadro: Desglose de Tareas: 3.2.2-.Aplicación Node.js-Depuración

<i>Paquete de trabajo:</i>	Aplicación Node.js-Depuración.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	4 horas/persona.
<i>Duración:</i>	4 horas.
<i>Descripción:</i>	Se realizarán tareas de optimización de código y arreglo de errores que se encuentren.
<i>Entradas:</i>	Implementación de la Base de Datos y de la lógica de negocio de la aplicación servidora Node.js
<i>Salidas/Entregables:</i>	Aplicación servidora Node.js sin errores y con un correcto funcionamiento.
<i>Recursos Necesarios:</i>	Editor HTML/Javascript, Conexión a la base de Datos MySQL y una persona.
<i>Precedencias:</i>	3.2.1-.Aplicación Node.js-Lógica de Negocio.

9. Cuadro: Desglose de Tareas: 3.3.1-.Aplicación Android-Interfaces

<i>Paquete de trabajo:</i>	Aplicación Android-Interfaces.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	5 horas/persona.
<i>Duración:</i>	5 horas.
<i>Descripción:</i>	Se implementará la interfaz de usuario para que el usuario pueda interactuar con la aplicación.
<i>Entradas:</i>	Prototipo de las pantallas que se van a utilizar en la interfaz.
<i>Salidas/Entregables:</i>	Implementación de la interfaz de la aplicación Android.
<i>Recursos Necesarios:</i>	IDE Android Studio y una persona.
<i>Precedencias:</i>	1.2-.Investigación Android, 3.2.1-.Aplicación Node.js-Lógica de Negocio.

10. Cuadro: Desglose de Tareas: 3.3.2-.Aplicación Android-Lógica de Negocio

<i>Paquete de trabajo:</i>	Aplicación Android-Lógica de Negocio.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	50 horas/persona.
<i>Duración:</i>	50 horas.
<i>Descripción:</i>	Se implementará la lógica de negocio para que la aplicación Android funcione,excluyendo la interfaz gráfica.
<i>Entradas:</i>	Diseño de diagramas de Secuencia.
<i>Salidas/Entregables:</i>	Lógica de Negocio de la aplicación Android.
<i>Recursos Necesarios:</i>	IDE Android Studio, aplicación servidora Node.js y una persona.
<i>Precedencias:</i>	1.2-.Investigación Android, 3.2.1-.Aplicación Node.js-Lógica de Negocio.

11. Cuadro: Desglose de Tareas: 3.3.3-.Aplicación Android-Depuración

<i>Paquete de trabajo:</i>	Aplicación Android-Depuración.
<i>Responsable:</i>	Jon Ander Fontán
<i>Esfuerzo:</i>	10 horas/persona
<i>Duración:</i>	10 horas
<i>Descripción:</i>	Se realizaran tareas de optimización de código y arreglo de errores que se encuentren.
<i>Entradas:</i>	Implementación de la Base de Datos, interfaz y lógica de negocio de la aplicación Android
<i>Salidas/Entregables:</i>	Aplicación Android sin errores y con un correcto funcionamiento.
<i>Recursos Necesarios:</i>	IDE Android Studio, aplicación servidora Node.js y una persona
<i>Precedencias:</i>	3.3.1-.Aplicación Android-Interfaces, 3.3.2-.Aplicación Android-Lógica de Negocio.

12. Cuadro: Desglose de Tareas: 3.4.1-.Aplicación Web-Interfaces

<i>Paquete de trabajo:</i>	Aplicación Web-Interfaces.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	5 horas/persona.
<i>Duración:</i>	5 horas.
<i>Descripción:</i>	Se implementará la interfaz de usuario para que el usuario pueda interactuar con la aplicación.
<i>Entradas:</i>	Prototipo de las pantallas que se van a utilizar en la interfaz.
<i>Salidas/Entregables:</i>	Implementación de la interfaz de la aplicación Android.
<i>Recursos Necesarios:</i>	Editor HTML/Javascript, aplicación servidora Node.js y una persona.
<i>Precedencias:</i>	3.2.1-.Aplicación Node.js-Lógica de Negocio.

13. Cuadro: Desglose de Tareas: 3.4.2-.Aplicación Web-Lógica de Negocio

<i>Paquete de trabajo:</i>	Aplicación Web-Lógica de Negocio.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	30 horas/persona.
<i>Duración:</i>	30 horas.
<i>Descripción:</i>	Se implementará la lógica de negocio para que la aplicación Web funcione, excluyendo la interfaz gráfica.
<i>Entradas:</i>	Diseño de diagramas de secuencia.
<i>Salidas/Entregables:</i>	Lógica de Negocio de la aplicación Web.
<i>Recursos Necesarios:</i>	Editor HTML/Javascript, aplicación servidora Node.js y una persona.
<i>Precedencias:</i>	3.2.1-.Aplicación Node.js-Lógica de Negocio.

14. Cuadro: Desglose de Tareas: 3.4.3-.Aplicación Web-Depuración

<i>Paquete de trabajo:</i>	Aplicación Web-Depuración.
<i>Responsable:</i>	Jon Ander Fontán.
<i>Esfuerzo:</i>	7.5 horas/persona.
<i>Duración:</i>	7.5 horas.
<i>Descripción:</i>	Se realizarán tareas de optimización de código y arreglo de errores que se encuentren.
<i>Entradas:</i>	Implementación de la Base de Datos, interfaz y lógica de negocio de la aplicación web.
<i>Salidas/Entregables:</i>	Aplicación Web sin errores y con un correcto funcionamiento.
<i>Recursos Necesarios:</i>	Editor HTML/Javascript, aplicación servidora Node.js y una persona.
<i>Precedencias:</i>	3.4.1-.Aplicación Web-Interfaces, 3.4.2-.Aplicación Web-Lógica de Negocio.

15. Cuadro: Desglose de Tareas: 4-.Manuales

<i>Paquete de trabajo:</i>	Manuales.
<i>Responsable:</i>	Jon Ander Fontán
<i>Esfuerzo:</i>	10 horas/persona
<i>Duración:</i>	10 horas
<i>Descripción:</i>	Realización de los manuales de usuario explicando el correcto uso de las aplicaciones.
<i>Entradas:</i>	Aplicaciones Android y Web, las cuales se va a explicar su funcionamiento.
<i>Salidas/Entregables:</i>	Manuales de usuario de las aplicaciones Android y Web
<i>Recursos Necesarios:</i>	Editor de documentos de látex y una persona.
<i>Precedencias:</i>	3.3.3-.Aplicación Android-Depuración, 3.4.3-.Aplicación Web-Depuración.

Herramientas

Para el desarrollo de este proyecto se han utilizado los siguientes programas:

- Android Studio

3. Figura: Logo Android Studio



Android Studio es utilizado para el desarrollo de la aplicación de Android *FastService*. Se eligió esta alternativa frente a eclipse, dado que los problemas que han surgido en el uso de ambos han sido menores en Android Studio (Pese a ser una beta).

- Sublime Text Editor

4. Figura: Logo Sublime Text Editor



Sublime Text Editor es utilizado tanto para el desarrollo de la aplicación web como para el desarrollo de la aplicación servidora Node.js. Es un editor que soporta amplia cantidad de lenguajes de programación y resulta muy cómodo de usar.

- Lyx

5. Figura: Logo Lyx



Lyx es el programa elegido para realizar la documentación. Es un editor de documentos que permite la inserción de código $LaTeX$. Este programa sustituye a la alternativa inicial *TexStudio* que fue descartada debido a lo complejo que resultaba realizar la documentación completa con una herramienta en la que tenía poca práctica.

- PHPMyAdmin

6. Figura: Logo PHPMyAdmin



PhpMyAdmin ha sido el entorno elegido para crear y manejar la base de datos y tiene como base un servidor web *Apache*.

- Microsoft Visio 2010 Professional

7. Figura: Logo Visio



Microsoft Visio 2010 ha sido uno de los programas elegidos para la realización de los diagramas del proyecto. En el se desarrollan los diagramas de Gantt y de secuencia dado que resulta una herramienta muy útil y proporciona diversas facilidades a la hora de crear los diagramas.

- Cacao

8. Figura: Logo Cacao



Cacao es una herramienta de diseño online, la cual resulta una potente herramienta de realización de diagramas y de bocetos de las aplicaciones.

- Dropbox

9. Figura: Logo Dropbox



Dropbox ha sido la herramienta utilizada para el almacenamiento de copias de respaldo en la nube. También ha sido elegida por el sencillo control de versiones que tiene, para poder recuperar archivos, más que suficiente para el desarrollo, si solo actúa una sola persona sobre los ficheros.

Gestión de Riesgos

A continuación se detallarán cada uno de los riesgos a los cuales podría ser vulnerable el sistema desarrollado, describiendo cada una de sus características y soluciones.

Para realizar este análisis, se impondrán unas ponderaciones a la probabilidad y el impacto:

1. Probabilidad

- Improbable [0-25 %].
- Poco Probable [25-50 %].
- Probable [50-75 %].
- Muy Probable [75-100 %].

2. Impacto

- Bajo.
- Medio.
- Alto.

16. Cuadro: Riesgo 1

Descripción	Retraso en el desarrollo del proyecto debido a la falta de experiencia en cuanto a la planificación.
Prevención	Añadir una mayor holgura en las tareas que no se tenga certeza del tiempo que llevaran.
Plan de contingencia	Realizar horas extras en la semana para no retrasar el resto de tareas.
Probabilidad	Probable.
Impacto	Alto.

17. Cuadro: Riesgo 2

Descripción	Retraso del desarrollo del proyecto debido a un accidente o una enfermedad.
Prevención	-
Plan de contingencia	Se redistribuye la carga de trabajo entre el resto de semanas restantes, lo que significaría un aumento de las horas que se dedica al proyecto en cada semana.
Probabilidad	Poco Probable.
Impacto	Medio.

18. Cuadro: Riesgo 3

Descripción	Localización de errores de programación en un estado muy avanzado del proyecto.
Prevención	Ir realizando pruebas de los distintos puntos que se van desarrollando.
Plan de contingencia	Ampliar el número de horas en las semanas restantes para la solución del error.
Probabilidad	Poco Probable.
Impacto	Alto.

19. Cuadro: Riesgo 4

Descripción	Perdida de Equipo de trabajo debido a accidentes (Hardware y Software)
Prevención	Guardar la información del proyecto en varias localizaciones como puede ser, discos externos o la nube, y no solo de forma local. Del mismo modo tener un equipo de reserva siempre que sea posible.
Plan de contingencia	Utilizar un equipo de sustitución, instalar el software necesario y recuperar los datos de las copias de respaldo.
Probabilidad	Improbable
Impacto	Alto

20. Cuadro: Riesgo 5

Descripción	Retraso en el proyecto debido a colapsos a la hora de desarrollar las distintas aplicaciones.
Prevención	Leer bien las documentaciones existentes de las aplicaciones para que no exista duda de como utilizar los distintos frameworks o complementos.
Plan de contingencia	Revisar bien el código en busca del error y en caso de que no se encuentre volver a un punto en el cual la aplicación funcionaba y seguir a partir de allí.
Probabilidad	Probable
Impacto	Medio

Evaluación Económica

Se ha realizado una evaluación económica, con la previsión del tiempo que se le va a dedicar al proyecto, se ha identificado el número de horas total a dedicar en cada tarea por semana.

Aunque en el desarrollo de este proyecto no se percibirá ningún ingreso, para realizar este apartado se ha tomado la decisión que el coste por hora/persona será de 10€.

21. Cuadro: Horas Semanales/Tarea

Tarea/Semana	1	2	3	4	5	6	7	8	9
Documentación	20								
Investigación Node.js	5								
Investigación Android		5							
Diseño de Base de Datos		1.5							
Diseño de Diagramas		18.5	11.5						
Desarrollo de la Base de Datos			2						
Aplicación Node.js-Lógica de Negocio			11.5	3.5					
Aplicación Node.js-Depuración				4					
Aplicación Android-Interfaces				5					
Aplicación Android-Lógica de Negocio				12.5	25	12.5			
Aplicación Android-Depuración						10			
Aplicación Web-Interfaces							5		
Aplicación Web-Lógica de Negocio							20	10	
Aplicación Web-Depuración								7.5	
Manuales								7.5	6.5

22. Cuadro: Desglose de Costes

Semana ->	1	2	3	4	5	6	7	8	9
Horas Asignadas	25	25	25	25	25	22.5	25	25	6.5
Coste Hora	10€	10€	10€	10€	10€	10€	10€	10€	10€
Ingresos	-	-	-	-	-	-	-	-	-
Coste Semanal	-250€	-250€	-250€	-250€	-250€	-225€	-250€	-250€	-65€
Coste Acumulado	-250€	-500€	-750€	-1000€	-1250€	-1475€	-1725€	-1975€	-2040€

Partiendo del supuesto de que no se percibirá ningún ingreso durante el desarrollo de este proyecto, el resultado obtenido en los cálculos realizados asciende a una pérdida de 2040€ por un trabajo realizado durante nueve semanas.

Antecedentes

Situación Actual

Actualmente en el campo de la hostelerías, concretamente en los restaurantes los camareros son el único medio a través del cual el cliente es capaz de realizar sus pedidos, suponiendo que el cliente este físicamente en dicho local. Sin embargo si un cliente quisiera realizar un pedido a distancia, tendría las opciones de: o bien realizar una llamada telefónica o utilizar una herramienta externa, como podría ser una web o una aplicación móvil.

Estudio de Diferentes Alternativas Existentes

En cuanto a las alternativas existentes, no hay ninguna que realice lo mismo que el sistema que se plantea en este proyecto, sin embargo existen dos formatos de aplicaciones, que podrían igualar funcionalidad, en caso de que este proyecto se quiera expandir a realizar pedidos a domicilio.

- Aplicaciones de pedidos a domicilio.

Existen distintas alternativas a la realización de pedidos a restaurantes a través de los dispositivos móviles, pero todas ellas mantienen básicamente la misma funcionalidad. El usuario deberá elegir su zona de vivienda, y la aplicación le sugiere distintos restaurantes a su alrededor, una vez que el usuario selecciona un restaurante elige entre los artículos disponibles y realiza el pedido.

Algunas de las aplicaciones más conocidas que cumplen esta funcionalidad son las siguientes:

- PedidosYa
 - (<https://play.google.com/store/apps/details?id=com.pedidosya&hl=es>)
- JUST EAT
 - (<https://play.google.com/store/apps/details?id=com.justeat.app.es&hl=es>)
- La Nevera Roja
 - (<https://play.google.com/store/apps/details?id=com.laneveraroja&hl=es>)

- Aplicaciones estilo TPV.

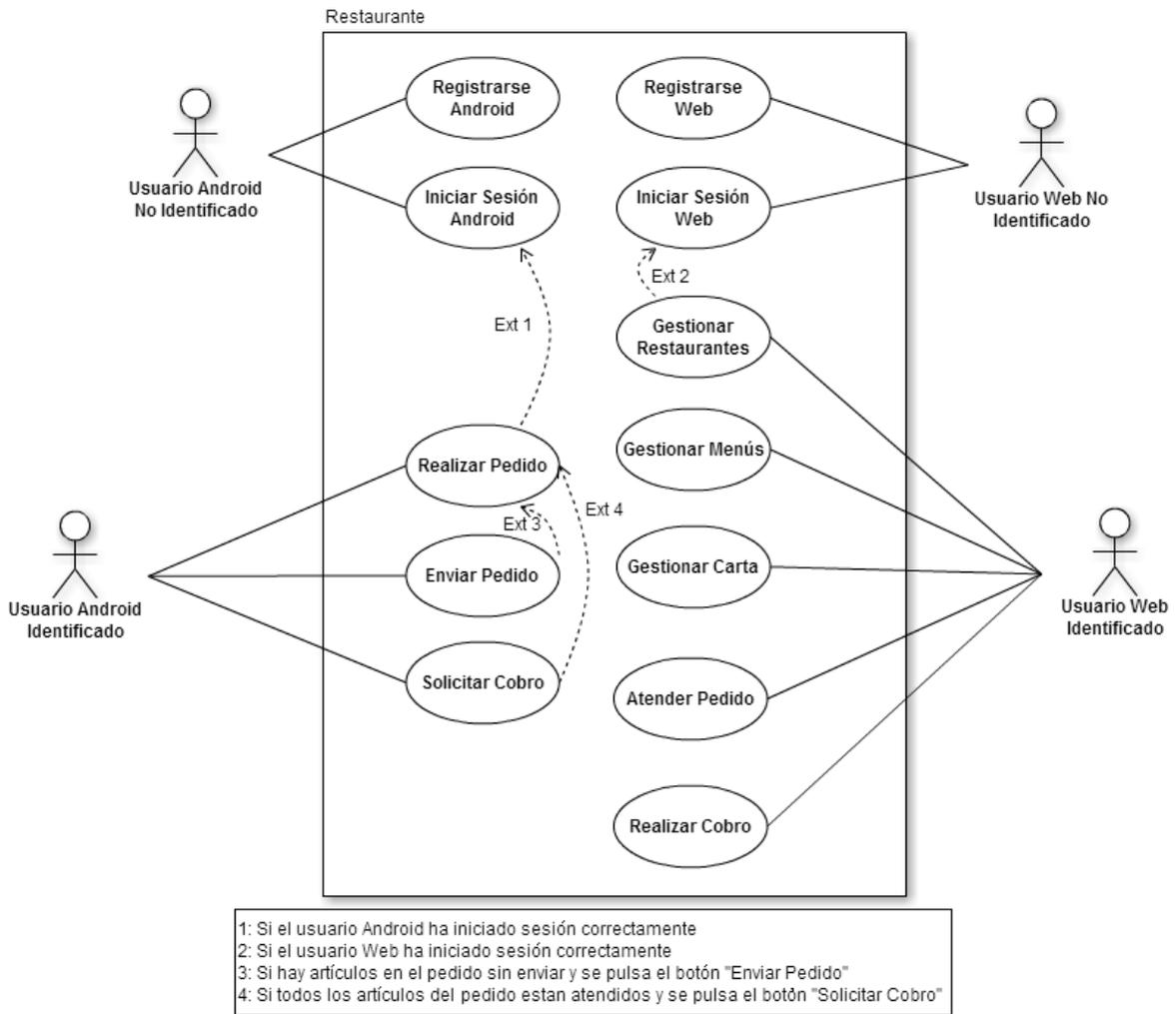
También existen aplicaciones que pueden facilitar la vida al camarero a la hora de tomar nota del pedido. Estas aplicaciones permiten al camarero ir con su dispositivo móvil e ir tomando apuntes de todo lo que piden los clientes.

- camarero 2.0 TPV Restaurante
 - (<https://play.google.com/store/apps/details?id=io.waiter.android&hl=es>)

Captura de Requisitos

Modelo de Casos de Uso

10. Figura: Casos de Uso



Esta figura representa únicamente el modelo de casos de uso, los casos de uso extendido se puede

Jerarquía de Actores

11. Figura: Jerarquía de Actores



En este caso disponemos de cuatro actores, dos por cada aplicación desarrollada. En cada una de las aplicaciones tenemos los actores de *Usuario Identificado* y *Usuario No Identificado*, el *Usuario No Identificado* podrá acceder a sus respectivas funcionalidades de Iniciar Sesión y Registrarse y el *Usuario Identificado* podrá hacer uso de todas las funciones disponibles en su aplicación.

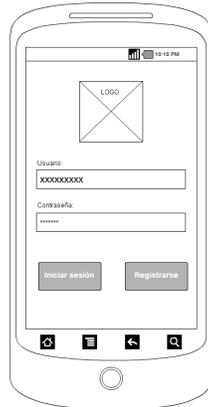
Prototipos de Interfaces de Usuario

Para realizar los prototipos me he basado en las líneas de diseño que Google ha puesto a disposición de los desarrolladores, que marcan como deberían ser los diseños para las aplicaciones Android.

El prototipo de la aplicación son en realidad dos prototipos, dado que la aplicación consta de dos partes una *aplicación Android* donde será el usuario el que interactúa y una *web de administración* donde el encargado de atender los pedidos y modificar todos los datos referentes al restaurante será el camarero.

- Aplicación Android

12. Figura: Boceto Android Inicio Sesión



En un primer momento, al iniciar la aplicación, lo que nos aparecerá es una ventana en la cual podemos ingresar nuestros datos de inicio de sesión, así como el logo de la aplicación como presentación de la misma, en caso de que no estemos registrados el botón “Registrarse” nos facilitará el acceso a la pantalla de registro.

13. Figura: Boceto Android Registro



La pantalla de registro contará con un formulario en el que se deberán rellenar los distintos campos que componen la información de usuario, como pueden ser: Usuario, Nombre, Contraseña y Correo Electrónico. Cuando el usuario haya completado el formulario y pulse en el botón “Registrarse”, se guardará la información en el sistema y se le llevara al usuario a la pantalla de Inicio de Sesión para que proceda a utilizar su cuenta.

14. Figura: Boceto Android Pedido



Una vez el usuario haya iniciado sesión y leído el código QR que se encuentra en la mesa, accederá a la pantalla de pedido. Esta pantalla será la principal, dado que desde esta podremos visualizar el estado de los elementos que se añadan al pedido, poder eliminarlos o editarlos, así como el coste total del pedido que lleva realizado hasta el momento. En la parte superior encontramos un botón que tendrá varias funciones, en caso de que tengamos artículos pendientes de enviar al restaurante nos dará la opción de enviarlos, por el contrario si todos los artículos del pedido se encuentran en el estado “Atendido” (Los artículos ya han sido tomados nota por parte del camarero y han sido servidos o se encuentran en proceso de preparación.) el botón tendrá como función la solicitud de cobro al camarero por parte del cliente.

En la parte inferior de la pantalla el usuario dispone de un botón para poder añadir más artículos a su pedido.

15. Figura: Boceto Android Menú



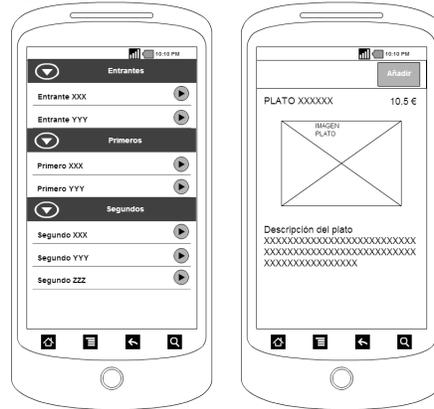
Si el usuario pulsa en el botón de menú de su dispositivo Android se le mostrará en forma de menú otras funciones como puede ser la visualización de su historial de pedidos en dicho restaurante (Si dispone de ella), la opción de eliminar el ticket en caso de que quiera abandonar el restaurante sin realizar un pedido (Esta opción solo se permitirá si el usuario no ha añadido ningún artículo al pedido), o la opción de cerrar la sesión en el dispositivo actual.

16. Figura: Boceto Android Pedido Popup



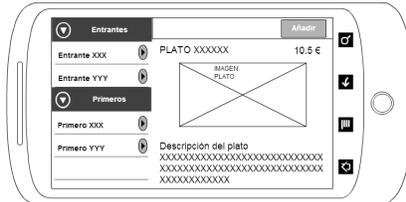
Si el usuario pulsa el botón del "Añadir" en el menú, aparecerá una ventana emergente que le pregunta que tipo de artículo desea añadir, "Artículo de la carta" o "Menú".

17. Figura: Boceto Android Selección Artículos



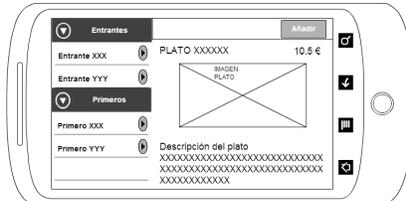
Cuando un usuario este añadiendo artículos a la carta, esta se mostrará en forma de lista desplegable (separada por tipo de plato) en la cual se ira visualizando el nombre de todos los artículos disponibles. En caso de seleccionar uno aparecerá una ventana con toda la información del artículo incluyendo el precio, una descripción del producto y una imagen de este. Desde esta ventana podremos añadir el artículo a la carta, lo que nos devolvería a la lista de artículos del restaurante.

18. Figura: Boceto Android Selección Artículos (Horizontal)



En caso de que se este añadiendo un menú lo que se hace es ir mostrando secuencialmente los platos de los distintos tipos. Es decir, Primero se muestran los primeros, una vez que se seleccione uno se mostrarán los segundos y así hasta que el usuario haya elegido el menú completo.

19. Figura: Boceto Android Selección Artículos (Horizontal)



En el caso de que el dispositivo este en modo horizontal, se mostrará a la izquierda la lista de los artículos y a la derecha se ira visualizando cada uno de ellos de la misma forma y con la misma funcionalidad que en el modo vertical. Esto se hará así para aprovechar el espacio de la pantalla en modo horizontal.

20. Figura: Boceto Android Historial



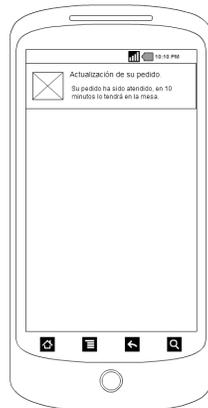
La visualización del Historial mantendrá una estética parecida a la de selección de artículos, en un primer momento se pondrá a disposición del usuario una lista de los distintos pedidos que ha realizado a dicho restaurante y si selecciona uno se visualizaran los detalles de este pedido en una ventana nueva.

21. Figura: Boceto Android Historial (Horizontal)



El modo horizontal también estará disponible en la visualización de los pedidos para poder aprovechar mejor la pantalla, mostrando en el lado izquierdo la lista del historial de pedidos y a la derecha los detalles del que se vaya seleccionando.

22. Figura: Boceto Android Notificación



En el momento que el usuario reciba una notificación por parte del sistema se mostrará de la forma nativa incluida en Android, se mostrará a la izquierda el logo de la aplicación, y a la derecha

un título avisándonos que nuestro pedido ha sido actualizado y a continuación el mensaje que nos transmite el sistema, ya sea que el pedido ha sido atendido por parte del camarero o que se ha procedido al pago del ticket.

- Web de administración

23. Figura: Boceto Web Iniciar Sesión

Web de FastService

Usuario:

Contraseña:

Al acceder a la web se nos mostrará un portal el cual nos permitirá iniciar sesión en el sistema de gestión de restaurantes, en ella el camarero deberá de facilitar el usuario y la contraseña del sistema para poder gestionar su restaurante.

En caso de que el restaurante no disponga de un usuario para gestionar su restaurante puede pulsar en el botón de registrarse para crear uno.

24. Figura: Boceto Web Registro

Usuario:

Nombre:

Contraseña:

Confirmar Contraseña:

Email:

La pantalla de registro contará con un formulario en el que se deberán rellenar los distintos campos que componen la información de usuario, como pueden ser: Usuario, Nombre, Contraseña y Correo Electrónico. Cuando el usuario haya completado el formulario y haya pulsado en el botón “Registrarse”, se guardará la información en el sistema y se le llevara al usuario a la pantalla de Inicio de sesión para que proceda a utilizar su cuenta.

25. Figura: Boceto Web Restaurante

Restaurante XXXXXXXXX
Descripción: XXXXXXXXXXXXXXXXXXXX

Mesa	Personas	Acciones
Mesa 0	4 Personas	<input type="button" value="Editar Mesa"/> <input type="button" value="Ver Código QR"/>
Mesa 1	6 Personas	<input type="button" value="Editar Mesa"/> <input type="button" value="Ver Código QR"/>

Nombre:

Descripción:

Logo:

La pantalla principal del restaurante visualizará el logo del restaurante así como el nombre que tiene y una pequeña descripción que el usuario le haya asignado, en la parte inferior se podrán gestionar las mesas disponibles por el restaurante, esto incluye el poder añadir mesas, editar las ya existentes o visualizar el código QR correspondiente a la mesa y el restaurante para poder ser imprimido y colocado en la mesa.

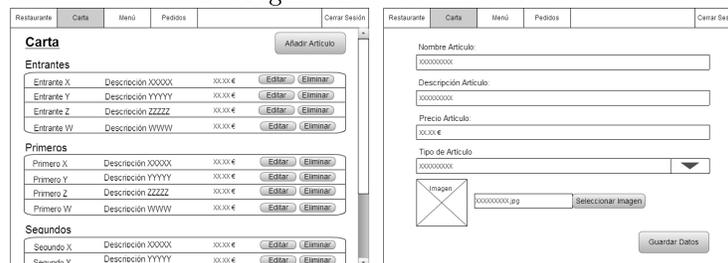
En caso de que el restaurante no haya facilitado la información del restaurante o haya pulsado sobre el botón de “Editar datos” se le mostrará una pantalla en la cual podrá ingresar todos los datos referentes al restaurante, como son; el nombre, una descripción y el logo.

26. Figura: Boceto Web Restaurante Popup



Si el usuario edita una mesa o añade una, aparecerá una ventana emergente la cual le permitirá ingresar el número de personas que puede albergar la nueva mesa. Una vez aceptado el número de personas, la nueva mesa aparecerá en la pantalla principal permitiendo al camarero interactuar con ella.

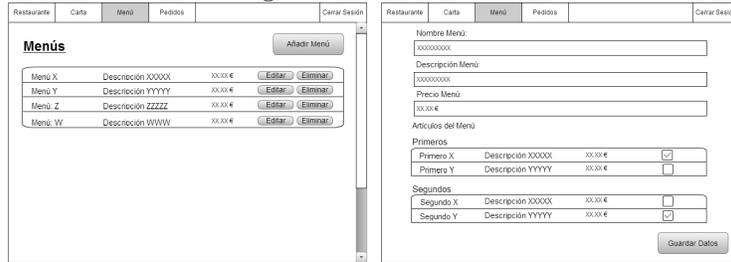
27. Figura: Boceto Web Carta



Si el usuario accede a la pantalla de gestión de la carta, lo que podrá visualizar es todos los artículos disponibles en la carta, separados por el tipo de plato que sea. En esta pantalla el usuario podrá añadir los distintos artículos a la carta así como modificar los ya existentes.

En caso de que el usuario pulse el botón de “Editar” o “Añadir Menú”, se mostrará una ventana con la información detallada de dicho artículo como es; el nombre del artículo, su descripción, el precio de dicho artículo, el tipo de plato que es y una imagen para hacerlo más vistoso. Si el usuario pulsa el botón de “Guardar Datos” volverá a la pantalla de Gestión de Artículos.

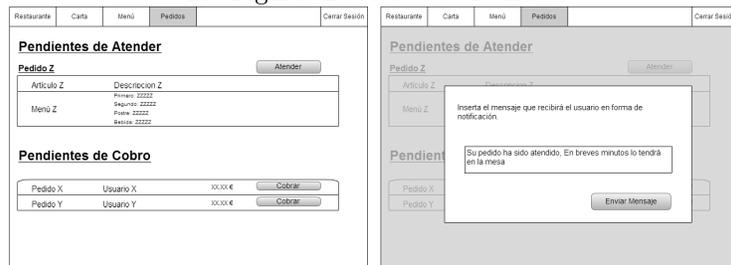
28. Figura: Boceto Web Menús



Si el usuario accede a la pantalla de gestión de los menús lo que podrá visualizar es todos los menús disponibles en el restaurante incluyendo su nombre, una descripción de cada uno y el precio.

En caso de que el usuario pulse el botón de “Editar” o “Añadir Menú”, se mostrará una ventana con la información detallada de dicho menú como es; el nombre del menú, su descripción, el precio de dicho artículo y una lista seleccionable de todos los artículos de la carta, de esta forma la selección de los artículos incluidos en el menú se realiza de una forma mas sencilla que una selección de una lista desplegable dado que se pueden ver todos de un vistazo. Si el usuario pulsa el botón de “Guardar Datos” volverá a la pantalla de Gestión de Menús.

29. Figura: Boceto Web Pedidos

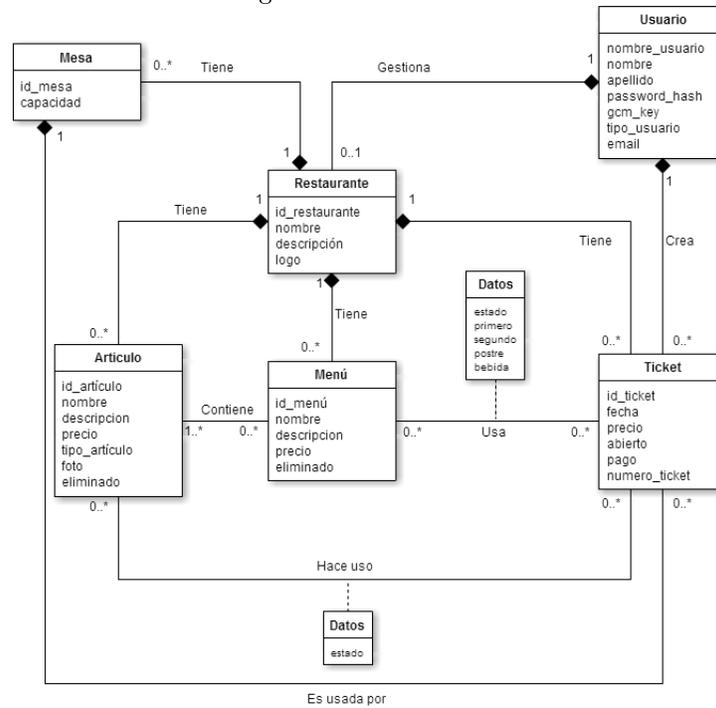


En la ventana de Gestión de pedidos, el camarero podrá visualizar por un lado todos los pedidos que los usuarios le han hecho llegar a través de la aplicación Android incluyendo la información de todos los artículos que el usuario ha incluido en el pedido. Si el camarero pulsa el botón “Atender” aparecerá una ventana emergente que le permitirá mandarle un mensaje personalizado al usuario indicando el tiempo aproximado de espera para que su pedido llegue a la mesa.

En la parte inferior de la pantalla de Gestión de Pedidos el usuario de la web podrá visualizar todas las solicitudes de cobro pendientes que se han llevado a cabo a través de la aplicación Android, en ellas se incluye el número de ticket, el nombre del usuario que ha realizado el pedido y el total de dinero. En el momento que el usuario haya cobrado al cliente la cantidad estimada en el ticket, deberá pulsar el botón “Cobrar” del ticket, esto cerrara el ticket y enviara un aviso al usuario del ticket avisándole de este evento.

Modelo de Dominio

30. Figura: Modelo de Dominio



Descripción de la Arquitectura

La Arquitectura de la aplicación, es una arquitectura de tres capas, dado que consideré más seguro que el cliente no pudiera acceder a la base de datos directamente, sino que se usara un servidor el cual se encargaría de realizar las llamadas a la base de datos y devolvería las respuestas al usuario.

31. Figura: Esquema de la Arquitectura del Sistema



Análisis y Diseño

Transformación de Modelo de Dominio a Base de Datos

Me he basado en el modelo de dominio del sistema para realizar las tablas y relaciones en la base de datos. A continuación se muestra el modelo relacional resultante.

32. Figura: Diagrama Relacional

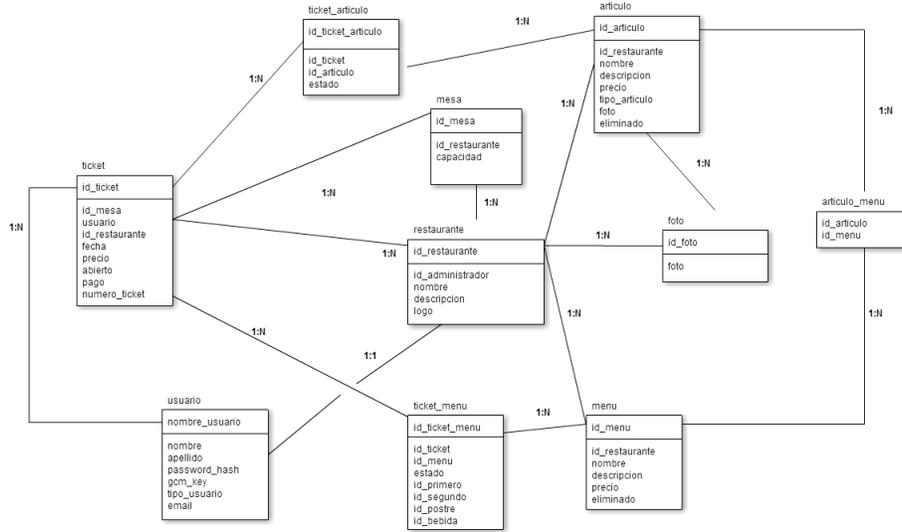
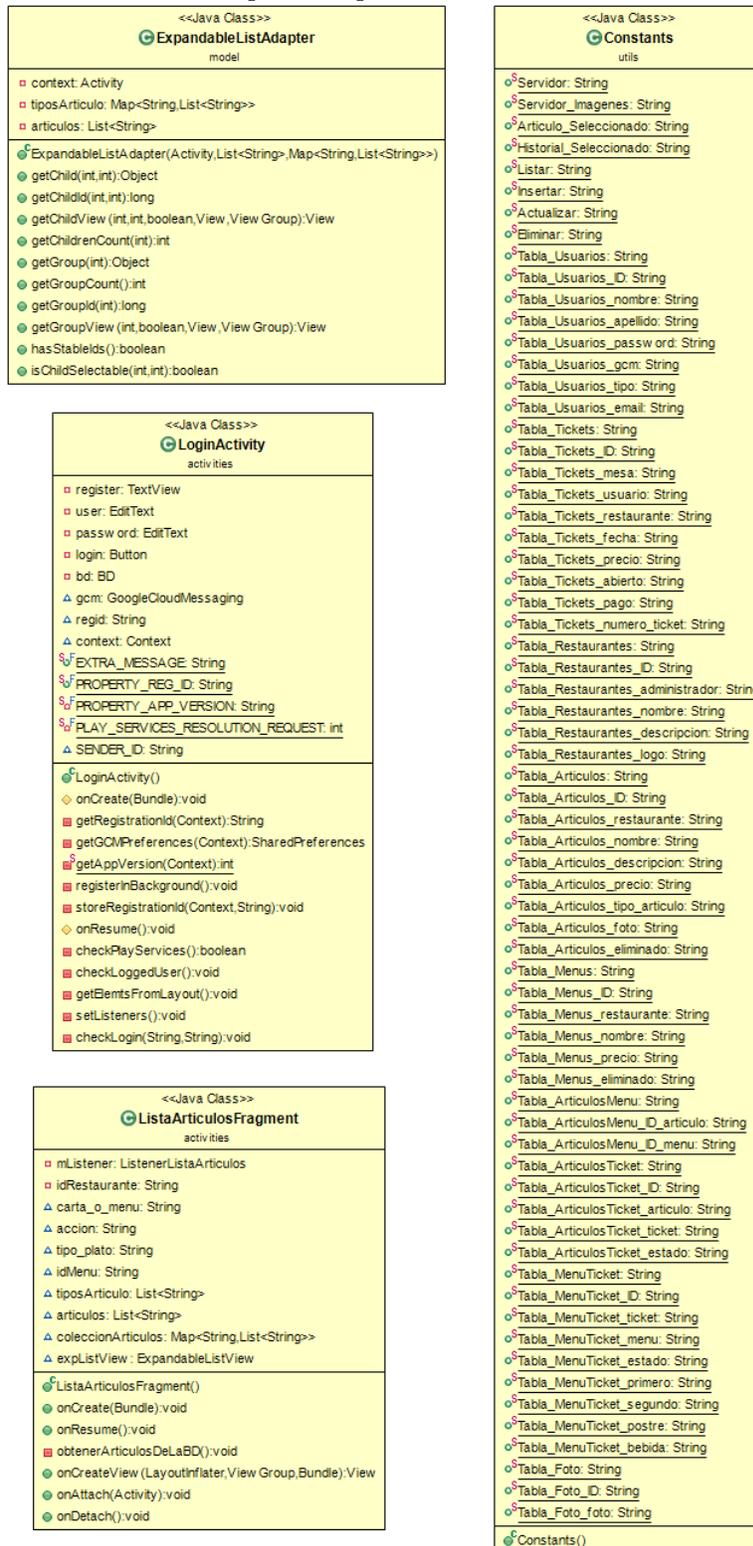
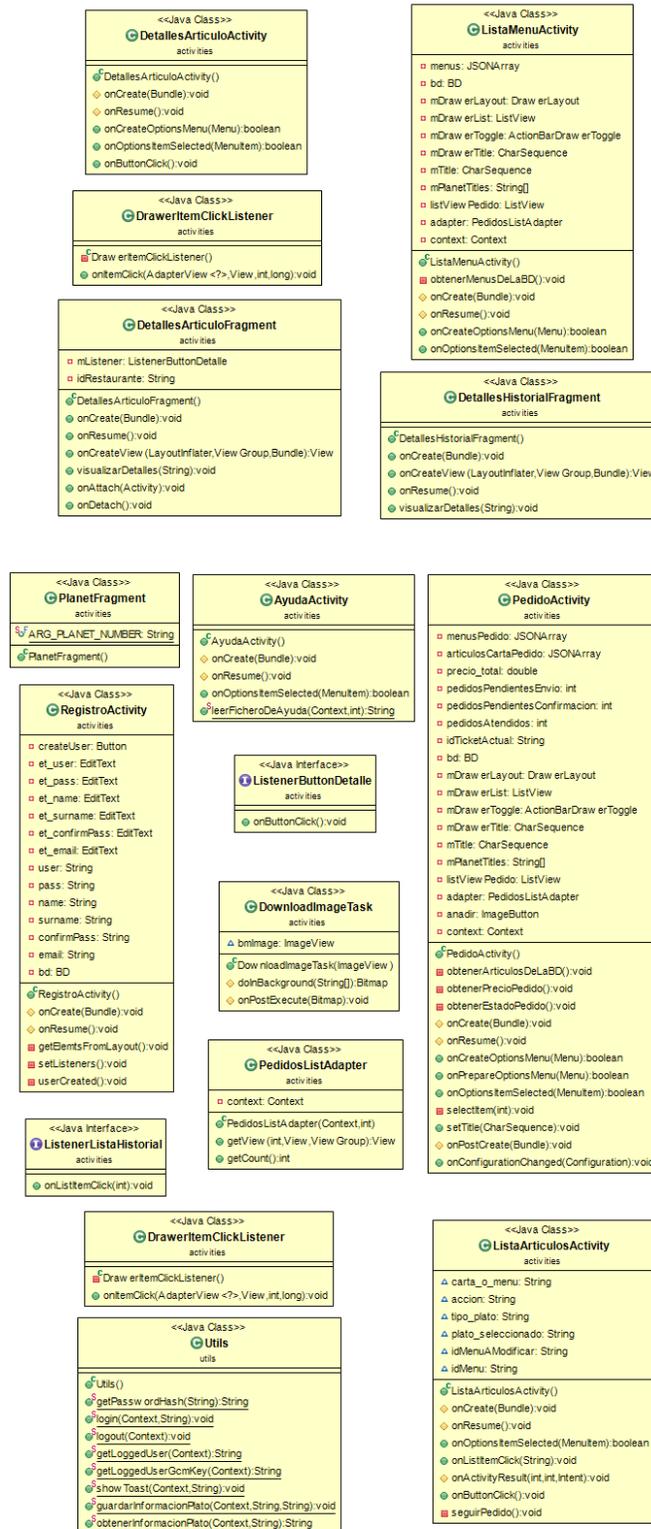


Diagrama de Clases

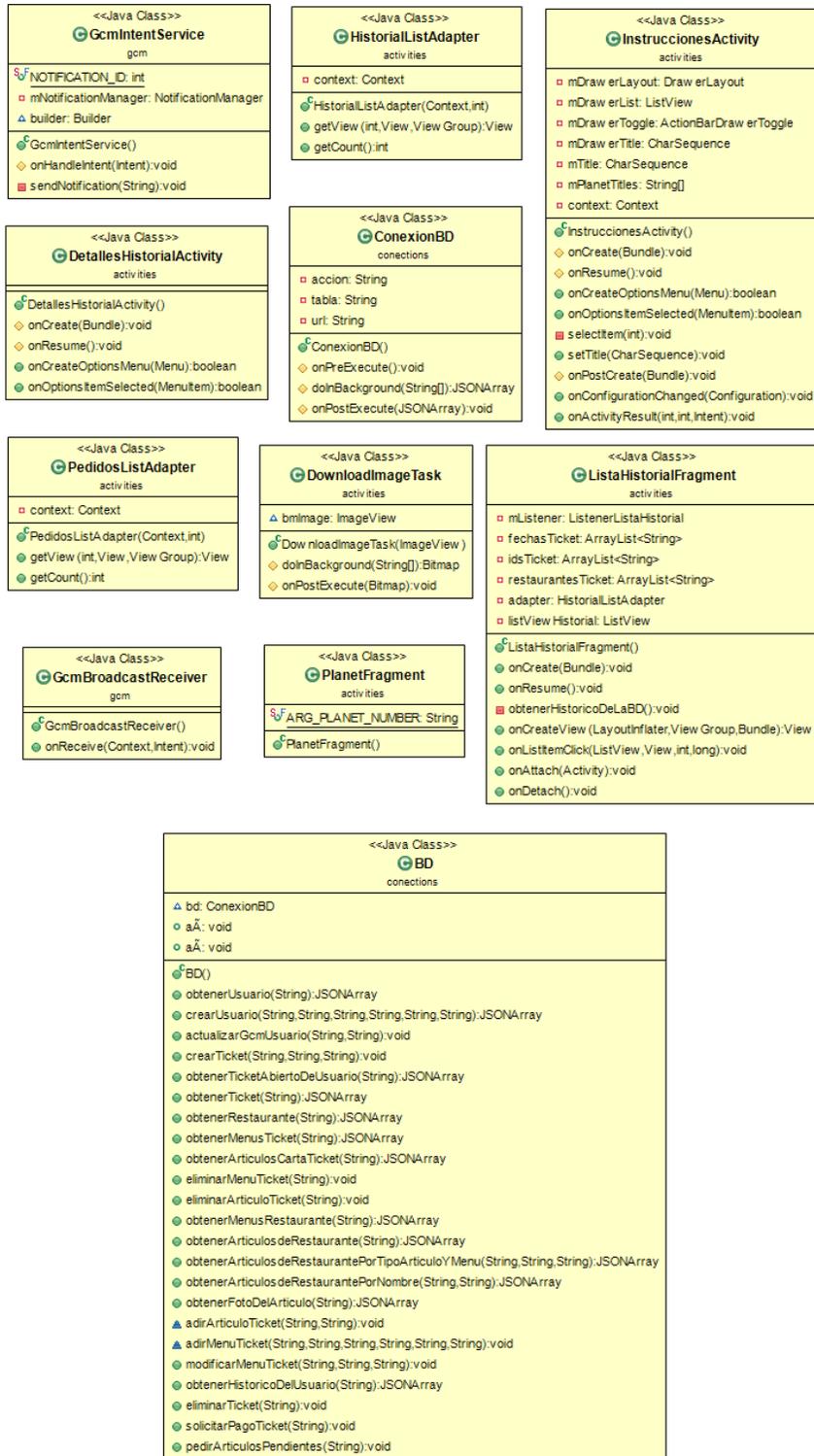
33. Figura: Diagrama de Clases Parte 1



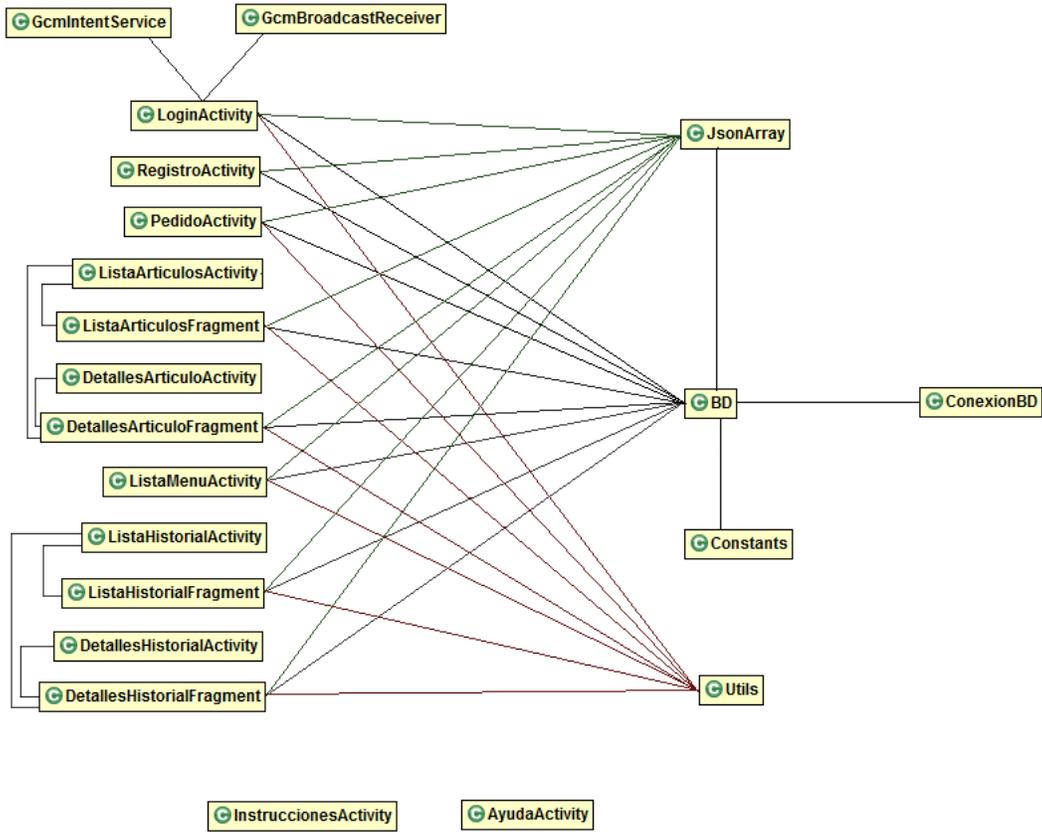
34. Figura: Diagrama de Clases Parte 2



35. Figura: Diagrama de Clases Parte 3

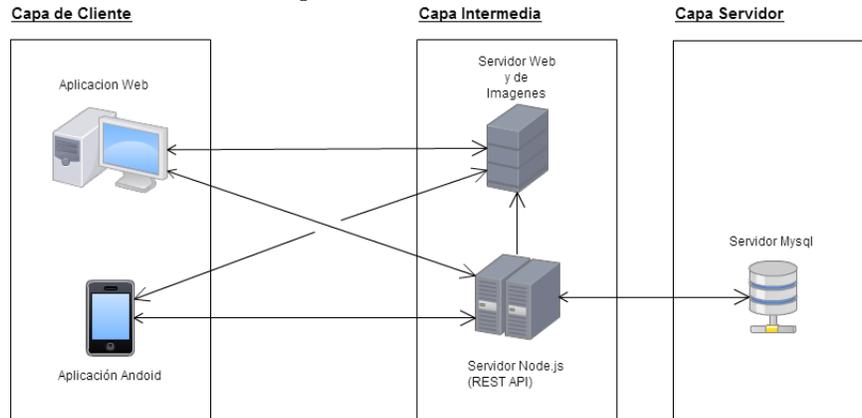


36. Figura: Diagrama de Clases Relaciones



Arquitectura del Sistema

37. Figura: Arquitectura del Sistema



■ Capa de Cliente

La capa de cliente es la que engloba tanto la aplicación web como la aplicación de Android. Tanto la aplicación web como la aplicación Android se comunican mediante peticiones HTTP al servidor Node.js para obtener los datos de la Base de Datos. A su vez ambas aplicaciones interactúan con el Servidor Web y de Imágenes pero por distintas causas, en el caso de la aplicación web realiza las peticiones tanto de paginas web como de las diferentes imágenes alojadas en este, al contrario que la aplicación Android que únicamente conecta para obtener las imágenes alojadas.

■ Capa Intermedia

La capa intermedia esta formada por dos servidores. El servidor web y de imágenes, que se encarga de atender las peticiones de la pagina web de FastService y de servir las imágenes que el sistema requiere, y el servidor de Node.js , que es el encargado de atender las peticiones de acceso a datos de la aplicación Web y la aplicación Android y facilitárselos.

El servidor web es un servidor Apache el cual es el encargado de alojar tanto las páginas de la web de administración, como las imágenes usadas por las aplicaciones.

El servidor Node.js fue elegido dado que ya consocia el uso de PHP como capa intermedia para el acceso a los datos y quería aprender el uso de JavaScript en el lado del servidor.

El servidor de Node.js interactúa con el servidor de imágenes en el momento que un usuario de la aplicación web quiere subir alguna imagen al sistema. Esta se almacena en una ruta y a su vez se guarda la referencia a esta en la base de datos.

■ Capa Servidor

Esta es la capa en la que se aloja la base de datos. En este caso es una base de datos MySQL dado que me proporcionaba la suficiente capacidad para realizar todas las tareas que yo necesitaba y porque estaba más habituado al uso de las bases de datos relacionales que a las no relacionales.

Diagrama de Secuencias

Los diagramas de secuencia se encuentran en el *Anexo II* que se puede encontrar en la parte final de esta documentación

Desarrollo

En este proyecto lo que se ha llevado a cabo es un sistema completo para la gestión de los restaurantes y las de sus correspondientes tickets.

Para llevar a cabo este objetivo se han realizado dos aplicaciones para que los usuarios puedan interactuar con el sistema.

- **Aplicación Android.**

Se trata de una aplicación a través de la cual los usuarios podrán realizar pedidos en el restaurante en el que se encuentren con la simple acción de leer el código que estará disponible en la mesa y eligiendo los artículos directamente desde su dispositivo móvil.

- **Aplicación Web**

Se trata de una aplicación que permitirá al usuario que este gestionando el restaurante, llevar el control completo de lo que los clientes son capaces de ver y adquirir mediante sus terminales. Desde la aplicación puede realizar todas las tareas rutinarias que requiere el sistema como pueden ser: gestionar la carta y los menús así como los datos del restaurante, y manejar los pedidos y cobros que los usuarios van generando.

- **Aplicación Servidora**

Se trata de una aplicación que hace de intermediario entre las aplicaciones del sistema y la base de datos. No hay más que un único programa que está accediendo a la base de datos el resto lo hacen a esta aplicación y se devuelven los datos correspondientes.

Estas aplicaciones han tenido un orden de desarrollo. Como es lógico la primera aplicación que se desarrolló fue la aplicación servidora, ya que sin ella las otras dos carecían de comunicación con la base de datos.

Durante el desarrollo de la aplicación servidora, tuve un tiempo de crisis al no conseguir que las imágenes se almacenaran correctamente en la base de datos y poder acceder a ellas de una forma correcta mediante Node.js. Ante esta situación se pesaron dos opciones, o bien rehacer la aplicación servidora por un servidor en PHP que si me lo permitía aunque con algunos cambios, o bien invertir más tiempo en la aplicación existente. La decisión fue tomada en base al tiempo de desarrollo que ya había invertido en la aplicación servidora. Al final se guardó mediante la aplicación las imágenes en otra localización (accesible desde el usuario) y guardaba en la base de datos una referencia a la misma.

En el momento de desarrollar las aplicaciones Web y Android surgieron errores y fallos en algunos puntos de la aplicación servidora lo que significó parar el desarrollo de las aplicaciones de usuario para retomar la aplicación servidora. Estos retrasos fueron acumulando un retraso en la planificación el cual se tuvo que solventar invirtiendo más horas de trabajo.

En el momento que las aplicaciones estuvieron desarrolladas y previamente probadas por mí, seleccioné a algunas personas de mi entorno que según probaron la aplicación realizaban acciones que yo no había probado lo que significó que hubo que recoger todos esos errores de programación, y solucionarlos antes de seguir con más pruebas dado que si en ese punto se hubieran encontrado más fallos arreglar luego dados los fallos de golpe y luego otra prueba con todas las cosas hubiera resultado más costoso para el usuario.

Verificación y Evaluación

Pruebas del Software

Pruebas de Aceptación

Durante el desarrollo de este proyecto las pruebas que se han llevado a cabo son las pruebas de aceptación.

Estas pruebas han sido llevadas a cabo en distintas fases:

- Fase Alfa

Durante esta fase solo el desarrollador ha sido el encargado de probar el programa, las pruebas se iban sucediendo en base al trabajo finalizado en cada momento, es decir, cada parte que se iba desarrollando se iba probando.

Durante esta fase eran muy comunes las reescrituras de código por problemas a la hora de programar no tenidos en cuenta.

- Fase Beta I

En esta fase se eligieron sujetos con conocimientos avanzados del uso de Android para que probaran la aplicación.

Una vez se encontraba algún fallo el Beta-Tester se lo comunicaba al desarrollador y este lo reparaba para una siguiente versión de la beta.

- Fase Beta II

En esta versión los sujetos elegidos para la prueba fueron ampliados y elegidos con un menor nivel de uso del sistema Android.

De esta manera tenía experiencias, tanto de usuarios avanzados como de usuarios principiantes.

La reparación de los errores encontrados se sucedió de la misma forma que en la Fase Beta I.

Documentación de Pruebas

Debido a la gran cantidad de pruebas realizadas para probar el correcto funcionamiento de las aplicaciones del sistema, a continuación se citan algunas de las pruebas realizadas en la fase alfa de las pruebas que es cuando más problemas surgieron a la hora de probar que todo funcionase.

23. Cuadro: Documentación de Pruebas Android

Descripción	Resultado Esperado	Resultado Obtenido
Registro de usuario con un nombre de usuario disponible.	Mensaje de aviso y se desplazará a la ventana de inicio de sesión.	Mensaje de aviso y se desplazará a la ventana de inicio de sesión.
Registro de usuario con un nombre de usuario no disponible.	Mensaje de error.	Mensaje de error.
Inicio de sesión de usuario con un nombre de usuario correcto y contraseña correcta.	Mensaje de aviso y desplazamiento hasta la pantalla de instrucciones.	Mensaje de aviso y desplazamiento hasta la pantalla de instrucciones.
Inicio de sesión de usuario con un nombre de usuario correcto y contraseña incorrecta.	Mensaje de error.	Mensaje de error.
Inicio de sesión de usuario con un nombre de usuario incorrecto y contraseña correcta.	Mensaje de error.	Mensaje de error.
Inicio de sesión de usuario con un nombre de usuario incorrecto y contraseña incorrecta.	Mensaje de error.	Mensaje de error.
Añadir artículo al pedido.	El artículo se añade al pedido.	El artículo se añade al pedido.
Eliminar artículo al pedido.	Aparece un mensaje de aviso para confirmar o denegar la acción. Si se ha pulsado si, se elimina el artículo. Si se ha pulsado no, se mantendrá el artículo.	Aparece un mensaje de aviso para confirmar o denegar la acción. Si se ha pulsado si, se elimina el artículo. Si se ha pulsado no, se mantendrá el artículo.
Añadir menú al pedido.	El menú se añade al pedido.	El menú se añade al pedido.
Añadir menú al pedido sin seleccionar ningún plato.	Se añade el menú al pedido, con cada uno de los platos en blanco.	Se añade el menú al pedido, con cada uno de los platos en blanco.
Añadir menú al pedido sin seleccionar algún plato.	Se añade el menú al pedido, con cada uno de los platos y dejará en blanco los no seleccionados.	Se añade el menú al pedido, con cada uno de los platos y dejará en blanco los no seleccionados.
Modificar menú al pedido.	Aparece un mensaje de aviso para saber que plato se modificará.	Aparece un mensaje de aviso para saber que plato se modificará.
Eliminar menú al pedido.	Aparece un mensaje de aviso para confirmar o denegar la acción. Si se ha pulsado si, se elimina el menú. Si se ha pulsado no, se mantendrá el menú.	Aparece un mensaje de aviso para confirmar o denegar la acción. Si se ha pulsado si, se elimina el menú. Si se ha pulsado no, se mantendrá el menú.
Envío de pedido con un artículo.	El pedido se añade correctamente a la base de datos.	El pedido se añade a la base de datos pero su número de ticket es 1 siempre.
Envío de pedido con un menú.	El pedido se añade correctamente a la base de datos.	El pedido se añade a la base de datos pero su número de ticket es 1 siempre.
Envío de pedido con variedad de artículos y menús.	El pedido se añade correctamente a la base de datos.	El pedido se añade a la base de datos pero su número de ticket es 1 siempre.
Solicitud de cobro de un pedido.	El pedido debería de aparecer como pendiente.	El pedido aparece como «pendiente» (con comillas) y no aparece en la web.

24. Cuadro: Documentación de Pruebas Web

Descripción	Resultado Esperado	Resultado Obtenido
Registro de usuario con un nombre de usuario disponible.	Mensaje de aviso y se desplazará a la ventana de inicio de sesión.	Mensaje de aviso y se desplazará a la ventana de inicio de sesión.
Registro de usuario con un nombre de usuario no disponible.	Mensaje de error.	Mensaje de error.
Inicio de sesión de usuario con un nombre de usuario correcto y contraseña correcta.	Mensaje de aviso y desplazamiento hasta la pantalla de instrucciones.	Mensaje de aviso y desplazamiento hasta la pantalla de instrucciones.
Inicio de sesión de usuario con un nombre de usuario correcto y contraseña incorrecta.	Mensaje de error.	Mensaje de error.
Inicio de sesión de usuario con un nombre de usuario incorrecto y contraseña correcta.	Mensaje de error.	Mensaje de error.
Inicio de sesión de usuario con un nombre de usuario incorrecto y contraseña incorrecta.	Mensaje de error.	Mensaje de error.
Atender pedido de usuario.	Los artículos del ticket se marcan como atendidos y se actualiza el precio del ticket.	Los artículos del ticket se marcan como atendidos y se actualiza el precio del ticket.
Cobrar el pedido del usuario.	El ticket cambia de estado a cerrado y se marca como pagado.	El estado es 'cerrado' (con comillas) lo cual no cerraba el ticket y el usuario seguía visualizando en la aplicación.
Añadir datos restaurante.	Los datos del restaurante se guardan correctamente.	Los datos del restaurante se guardan correctamente.
Añadir mesa al restaurante.	Se añade una mesa al restaurante.	La mesa no se añade correctamente al restaurante (problema en el servidor).
Añadir mesa al restaurante.	Se añade una mesa al restaurante.	Se añade una mesa al restaurante.
Modificar mesa al restaurante.	Se permitirá cambiar la capacidad de la mesa.	Se permite cambiar la capacidad de la mesa.
Añadir artículo a la carta con nombre no existente.	El artículo se guarda correctamente en la base de datos.	La imagen del artículo no se guarda correctamente.
Añadir artículo a la carta con nombre existente.	Mensaje de error.	Mensaje de error.
Modificar artículo de la carta	Aparecerá la pantalla con los datos para ser editados.	Aparece la pantalla con los datos para ser editados.
Eliminar artículo de la carta	Aparecerá la carta del restaurante con el artículo eliminado.	Aparece la carta del restaurante con el artículo eliminado.
Añadir menú.	Se añade un menú correctamente así como los artículos seleccionados.	Los datos del menú son guardados pero guarda solo el primer artículo elegido.
Modificar menú sin artículos seleccionados	El menú se modifica en la BD y no contiene artículos.	El menú se modifica en la BD y no contiene artículos.
Modificar menú con artículos seleccionados	El menú se modifica en la BD y contiene artículos.	El menú se modifica en la BD pero no contiene artículos en el .
Eliminar menú.	Aparecerá el listado de menús del restaurante sin el menú.	Aparece el listado de menús del restaurante sin el menú.

Conclusiones y Trabajo Futuro

El trabajo realizado se ha ajustado a lo comprometido en la propuesta entregada al comienzo del mismo. Todos los hitos principales han sido alcanzados con éxito, no así los objetivos secundarios que no ha sido posible realizarlos por falta de tiempo dado que se trataba de un proyecto de muchos componentes.

El único cambio respecto a la propuesta inicial ha sido la utilización del programa *Lyx* en sustitución de *TexStudio* para realizar la documentación. Este cambio fue debido a la dificultad que entrañaba el uso de un editor de documentos que era 100% código *LaTeX* a pasar a uno que a pesar de que permite añadir los elementos de una manera más gráfica, también permite la inserción de código *LaTeX* para insertar elementos.

Creo que la idea puede tener un verdadero futuro, siempre que logre una aceptación entre el colectivo de la hostelería, sin embargo el proyecto no finaliza aquí. Existen múltiples caminos a seguir a partir de este punto, como puede ser el añadir funcionalidad de poder generar pedidos a domicilio y sin que exista la necesidad de estar físicamente en el restaurante. Este cambio podría significar una reestructuración de la base de datos dado que está asumiendo que el cliente está situado en una mesa, y el tiempo del desarrollo a dedicar también sería otro punto a tener en cuenta.

Otro punto a mejorar es la *web de administración*. Dado que al ser un complemento de la *aplicación Android*, no se le ha otorgado toda la funcionalidad que merece una web de administración. Se le podría añadir funcionalidades como la de ver el historial de pedidos de los clientes, visualización de la caja diaria y estadísticas de adquisición por artículo, entre otros. Realizar esta mejora requiere más tiempo de reuniones con los usuarios de los restaurantes más que de desarrollo dado que la forma en la que a cada cual le gusta visualizar los datos es muy subjetiva y muy difícil de conseguir en una primera versión.

La *aplicación servidora* en este momento está alojada en un servidor privado y de poca capacidad de procesamiento/almacenamiento (*Raspberry Pi*) para mejorar el rendimiento de los servidores sería adecuado migrar la base de datos y la aplicación servidora a un servidor de mayor capacidad.

Por último si tuviera que hacer otro proyecto de similares características, me esforzaría más en conseguir una *planificación del tiempo correcta* dado que aunque yo pensaba que los dos meses de desarrollo eran suficientes, en la recta final tuve que realizar más horas de trabajo debido al estrés de la fecha de entrega y los fallos de última hora que surgían.

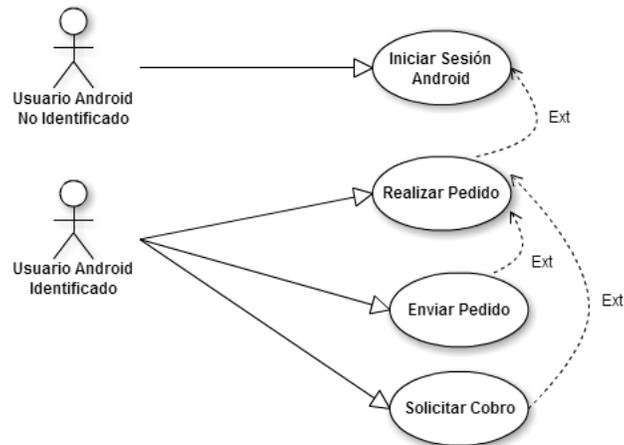
Bibliografía

- Google - Documentación oficial de Android (<http://developer.android.com/>)
- Joyent - Documentación oficial de Node.js (<http://nodejs.org/api/>)
- Jack Lukic - Documentación oficial Semantic UI - 2013 (<http://semantic-ui.com/>)
- W3CSchools - JavaScript Tutorial (<http://www.w3schools.com/js/>)
- Acharya Vaddey - Node.js MySQL Tutorial for beginners - 16 de Octubre del 2013 (<http://www.tutorialindustry.com/js-mysql-tutorial-for-beginners>)
- Marcus Farkas - GCM with Node.js - 2013 (<https://github.com/ToothlessGear/node-gcm>)
- StackOverflow - Consultas varias sobre problemas surgidos durante el desarrollo del proyecto. (<http://stackoverflow.com/>)
- Mikel Villamañe - Temario De la asignatura Análisis y Diseño de Sistemas Informáticos.
- Hector de la Presa - Temario de la asignatura Gestión de Proyectos.
- Juan Antonio Pereira - Desarrollo de Aplicaciones Web Enriquecidas.

ANEXO I.- Casos de Uso Extendidos

Caso de Uso Extendido Android: Iniciar Sesión

38. Figura: CDU Extendido Android: Iniciar Sesión



Descripción

El usuario que previamente se habrá registrado en el sistema deberá iniciar sesión para poder comenzar a usarlo.

Actores

Usuario Android no identificado.

Precondiciones

El usuario debe estar registrado en el sistema.

Flujo de eventos

1. Aparecen campos para completar con los datos del usuario (Figura: Inicio Sesión).
2. [Si el usuario pulsa el botón “Iniciar Sesión”]
 - 2.1. [Si los datos son correctos]
 - 2.1.1. Nos aparecerá una ventana para escanear el código QR.
 - 2.1.2. [Si se escanea correctamente]
 - 2.1.2.1. EXTENDS Realizar Pedido.
 - 2.1.3. [Si no se escanea correctamente]
 - 2.1.3.1. Se mostrará un mensaje de que no se ha escaneado correctamente y permitirá seguir escaneando.

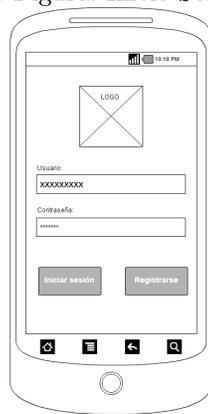
- 2.2. [Si los datos no son correctos]
- 2.2.1 Aparecerá un mensaje de error y el usuario podrá intentar rellenar los datos de nuevo.

Postcondiciones

El usuario habrá iniciado sesión y podrá realizar pedidos.

Imágenes

39. Figura: Inicio Sesión



Caso de Uso Extendido Android: Registrarse

40. Figura: CDU Extendido Android: Registrarse



Descripción

El usuario tendrá que registrarse en el sistema para poder acceder a realizar pedidos.

Actores

Usuario Android no identificado.

Precondiciones

El usuario ha pulsado el botón Registrar de la pantalla de inicio de sesión (Figura Inicio Sesión).

Flujo de eventos

- 1 Aparecen campos para completar con los datos del usuario (Figura Registro).
- 2 [Si el usuario pulsa el botón “Registrarse”]
 - 2.1 [Si el usuario No existe en el sistema]
 - 2.1.1 Nos aparecerá un mensaje de que se ha registrado correctamente y nos Lleva a la pantalla de iniciar sesión.
 - 2.2 [Si el usuario ya existe]
 - 2.2.1 Aparecerá un mensaje de error y el usuario podrá modificar los datos.

Postcondiciones

El usuario se habrá registrado en el sistema.

Imágenes

41. Figura: Inicio Sesión

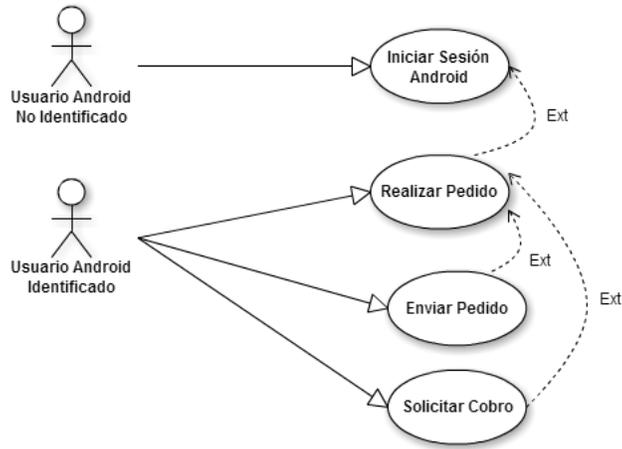


42. Figura: Registro



Caso de Uso Extendido Android: Realizar Pedido

43. Figura: CDU Extendido Android: Realizar Pedido



Descripción

El usuario añadir artículos a la lista para poder realizar el pedido.

Actores

Usuario Android identificado.

Precondiciones

El usuario ha escaneado correctamente el código QR.

Flujo de eventos

- 1 Una ventana en la cual el usuario podrá visualizar los artículos que tiene en su pedido (Inicialmente Ninguno) y los botones de Enviar Pedido, Solicitar Cobro y Añadir (Figura Pedido Vacío). Cuando se añadan elementos al pedido inicialmente podrán eliminarse y modificarse si el artículo es del tipo un menú (Figura Pedido Lleno).
- 2 [Si el usuario pulsa el botón Añadir]
 - 2.1 Aparece una ventana en la cual tiene que seleccionar si es un elemento de la carta o un menú (Figura Selección Artículo).
 - 2.2 [Si es pulsa en carta]
 - 2.2.1 Aparecerá una lista desplegable de los diferentes tipos de platos y dentro de cada una aparecen los platos correspondientes (Figura Lista Artículos).
 - 2.2.2 [Si el usuario selecciona un artículo]

- 2.2.2.1 Se mostrara el detalle del artículo que ha seleccionado. (Figura Detalle Artículo).
- 2.2.2.2 [Si el usuario pulsa en Añadir]
 - 2.2.2.2.1 El artículo se añadirá al pedido y se vuelve a mostrar la lista de todos los artículos de la carta. (Figura Lista Artículo).
- 2.2.2.3 [Si el usuario pulsa Atrás en su dispositivo]
 - 2.2.2.3.1 Se mostrará el resumen del pedido con los artículos que el usuario ha añadido. (Figura Pedido Lleno).
- 2.3 [Si pulsa en menú]
 - 2.3.1 Aparecerá una lista desplegable en la cual solo se mostrarán los artículos del tipo de plato que este seleccionando (Figura Lista Artículos Tipo Plato).
 - 2.3.2 [Si el usuario selecciona un artículo]
 - 2.3.2.1 Se mostrará el detalle del articulo que ha seleccionado (Figura Detalle Artículo).
 - 2.3.2.2 [Si el usuario pulsa en Añadir]
 - 2.3.2.2.1 [Si el plato que elije es el último del menú (bebida)]
 - 2.3.2.2.1.1 Se mostrará el resumen del pedido con el menú que el usuario ha añadido (Figura Pedido Lleno).
 - 2.3.2.2.2 [Si el plato que elije no es el último del menú (bebida)]
 - 2.3.2.2.2.1 Se mostrará la lista de artículos del siguiente tipo de plato a añadir al menú (Figura Lista Artículos Tipo Plato).
- 3 [Si el usuario pulsa Eliminar en algún elemento]
 - 3.1 Se eliminará el artículo del pedido y se mostrará la lista de artículos que el usuario tiene en el pedido (Figura Pedido Lleno).
- 4 [Si el usuario pulsa Editar en algún menú]
 - 4.1 Aparecerá una ventana emergente con los diferentes platos que puede editar en el menú (Figura Selección Articulo Menú).
 - 4.2 [Si el usuario selecciona algún plato]
 - 4.2.1 Aparecerá una lista desplegable en la cual solo se mostraran los artículos del tipo de plato que este seleccionando (Figura Lista Artículos Tipo Plato).
 - 4.2.2 [Si el usuario selecciona un articulo]
 - 4.2.2.1 Se mostrará el detalle del articulo que ha seleccionado. (Figura Detalle Artículo).
 - 4.2.2.2 [Si el usuario pulsa en Añadir]
 - 4.2.2.2.1 Se mostrará el resumen del pedido con el menú ya modificado. (Figura Pedido Lleno).

5 [Si el usuario pulsa el botón Enviar Pedido]

5.1 EXTENDS Enviar Pedido.

6 [Si el usuario pulsa el botón Solicitar Cobro]

6.1 EXTENDS Solicitar Cobro.

Postcondiciones

El usuario realizará un pedido al restaurante en el que se encuentra.

Imágenes

44. Figura: Pedido Vacío



45. Figura: Pedido Lleno



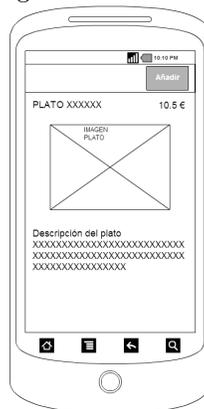
46. Figura: Selección Artículo



47. Figura: Lista Artículos



48. Figura: Detalle Artículo



49. Figura: Selección Artículo Menú

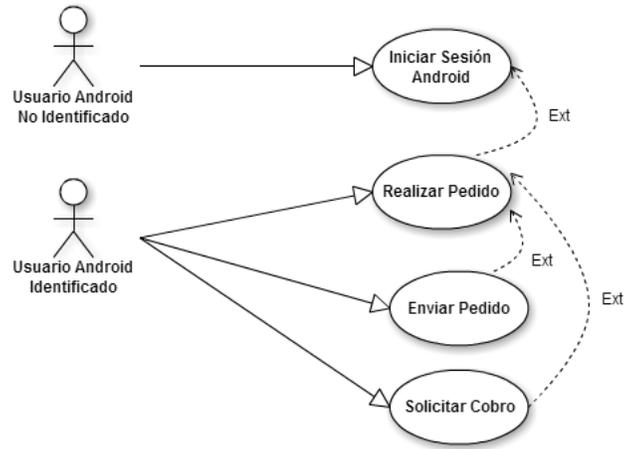


50. Figura: Lista Artículos Tipo Plato



Caso de Uso Extendido Android: Enviar Pedido

51. Figura: CDU Extendido Android: Enviar Pedido



Descripción

El usuario identificado después de haber realizado previamente la elección de sus artículos enviará su pedido para ser atendido por el restaurante.

Actores

Usuario Android identificado.

Precondiciones

El usuario se debe encontrar realizando un pedido.

Flujo de eventos

- 1 Se muestra una ventana con los artículos seleccionados con sus botones de "Editar" y "Borrar", el botón "Añadir", el botón "Enviar Pedido" y "Solicitar Cobro" (Figura Pedido).
- 2 [Si el usuario pulsa el botón "Enviar Pedido"]
 - 2.1 [Si alguno de los elementos esta en estado de pendiente de enviar]
 - 2.1.1 Se guardará la información de que los artículos pendientes han sido enviados y el usuario de la web lo podrá atender.
 - 2.2 [Si ninguno de los elementos esta en estado de pendiente de enviar]
 - 2.2.1 Se muestra un aviso de que no hay artículos pendientes que se puedan enviar.

Postcondiciones

Los elementos del pedido que no se habían enviado para ser atendidos ahora pasan al estado pendiente.

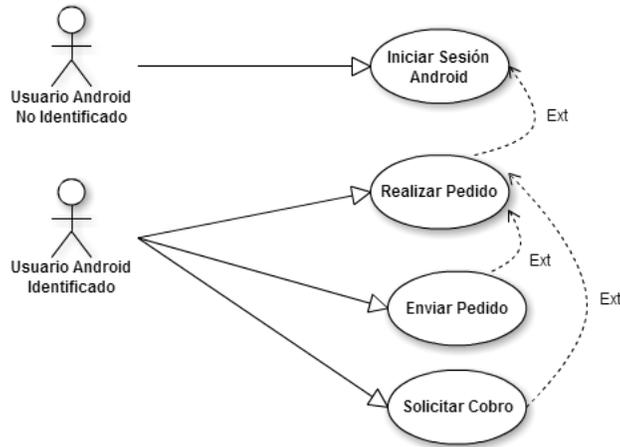
Imágenes

52. Figura: Pedido



Caso de Uso Extendido Android: Solicitar Cobro

53. Figura: CDU Extendido Android: Solicitar Cobro



Descripción

El usuario tendrá un pedido realizado el cual se dispondrá a solicitar el pago.

Actores

Usuario Android identificado.

Precondiciones

El usuario se debe encontrar realizando un pedido y haber enviado el pedido de todos los artículos que tenga en la lista.

Flujo de eventos

- 1 Se muestra una ventana con los artículos seleccionados con sus botones de "Editar" y "Borrar", el botón "Añadir", el botón "Enviar Pedido" y "Solicitar Cobro"(Figura Pedido).
- 2 [Si el usuario pulsa el botón "Solicitar Cobro"]
 - 2.1 [Si todos los artículos del pedido están en estado atendido]
 - 2.1.1 Se guardará la información de que el pedido esta listo para ser cobrado y el usuario de la web lo podrá atender.
 - 2.2 [Si todos los artículos del pedido no están en estado atendido]
 - 2.2.1 Se muestra un aviso de que todos los artículos no han sido atendidos.

Postcondiciones

El pedido realizado por el usuario estará pendiente de cobro.

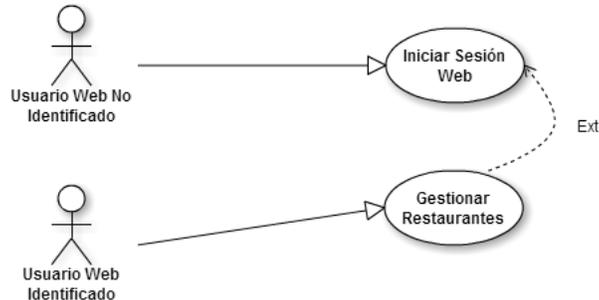
Imágenes

54. Figura: Pedido



Caso de Uso Extendido Web: Iniciar Sesión

55. Figura: CDU Extendido Web: Iniciar Sesión



Descripción

El usuario que previamente se habrá registrado en el sistema deberá iniciar sesión para poder comenzar a usarlo.

Actores

Usuario Web no identificado.

Precondiciones

El usuario debe estar registrado en el sistema.

Flujo de eventos

1. Aparecen campos para completar con los datos del usuario (Figura: Inicio Sesión).
2. [Si el usuario pulsa el botón “Iniciar Sesión”]
 - 2.1. [Si los datos son correctos]
 - 2.1.2.1. EXTENDS Gestionar Restaurante.
 - 2.2. [Si los datos no son correctos]
 - 2.2.1. Aparecerá un mensaje de error y el usuario podrá intentar rellenar los datos de nuevo.

Postcondiciones

El usuario habrá iniciado sesión.

Imágenes

56. Figura: Inicio Sesión



The image shows a login form for 'Web de FastService'. It features a logo placeholder (a square with an 'X' and the word 'Logo' inside) and the text 'Web de FastService'. Below this, there are two input fields: 'Usuario' and 'Contraseña'. At the bottom, there are two buttons: 'Iniciar Sesión' and 'Registrarse'.

Logo

Web de FastService

Usuario

Contraseña

Iniciar Sesión Registrarse

Caso de Uso Extendido Web: Registrarse

57. Figura: CDU Extendido Web: Registrarse



Descripción

El usuario tendrá que registrarse en el sistema para poder acceder a la gestión del restaurante.

Actores

Usuario Web no identificado.

Precondiciones

El usuario ha pulsado el botón Registrar de la pantalla de inicio de sesión (Figura Inicio Sesión).

Flujo de eventos

- 1 Aparecen campos para completar con los datos del usuario (Figura Registro).
- 2 [Si el usuario pulsa el botón “Registrarse”].
 - 2.1 [Si el usuario No existe en el sistema]
 - 2.1.1 Nos aparecerá un mensaje de que se ha registrado correctamente y nos llevará a la pantalla de iniciar sesión (Figura Inicio Sesión).
 - 2.2 [Si el usuario ya existe]
 - 2.2.1 Aparecerá un mensaje de error y el usuario podrá modificar los datos.

Postcondiciones

El usuario se habrá registrado en el sistema.

Imágenes

58. Figura: Inicio Sesión



The login form for 'Web de FastService' features a logo placeholder (a square with an 'X' and the word 'Logo' inside) and the text 'Web de FastService'. Below this, there are two input fields: 'Usuario' and 'Contraseña'. At the bottom, there are two buttons: 'Iniciar Sesión' and 'Registrarse'.

Logo

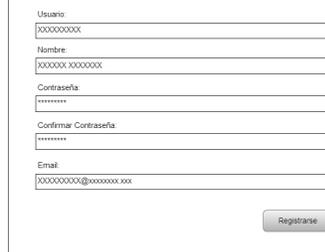
Web de FastService

Usuario

Contraseña

Iniciar Sesión Registrarse

59. Figura: Registro



The registration form contains several input fields: 'Usuario' (with placeholder 'XXXXXXXX'), 'Nombre' (with placeholder 'XXXXXXXX'), 'Contraseña' (with placeholder 'XXXXXXXX'), 'Confirmar Contraseña' (with placeholder 'XXXXXXXX'), and 'Email' (with placeholder 'XXXXXXXXX@XXXXXXXX.XXX'). A 'Registrarse' button is located at the bottom right.

Usuario

XXXXXXXX

Nombre

XXXXXXXX

Contraseña

XXXXXXXX

Confirmar Contraseña

XXXXXXXX

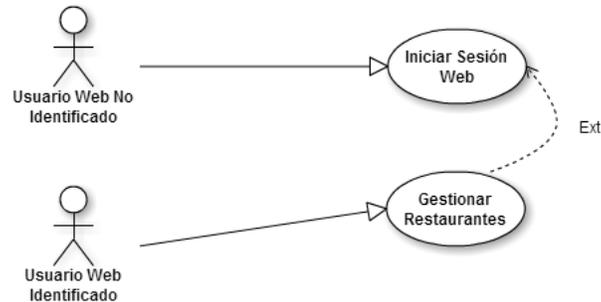
Email

XXXXXXXXX@XXXXXXXX.XXX

Registrarse

Caso de Uso Extendido Web: Gestionar Restaurante

60. Figura: CDU Extendido Web: Gestionar Restaurante



Descripción

El Usuario Identificado después de haber pulsado el botón “Restaurante” del menú superior tendrá acceso a la gestión de su restaurante.

Actores

Usuario Web Identificado

Precondiciones

El usuario deberá haber pulsado en el botón Restaurante del menú superior o que acabe de iniciar sesión.

Flujo de eventos

- 1 [Si el usuario no tiene creado su restaurante]
 - 1.1 Se cargará la página de edición del restaurante con sus campos completamente vacíos (Figura Restaurante 2).
 - 1.2 [Si el usuario pulsa el botón “Guardar Datos”]
 - 1.2.1 [Si todos los datos del formulario han sido rellenados]
 - 1.2.1.1 Se procede al guardado de los datos.
 - 1.2.1.2 Se mostrará la pantalla de información del restaurante con todos los datos (Figura Restaurante 1).
 - 1.2.2 [Si todos los datos del formulario no han sido rellenados]
 - 1.2.2.1 Se mostrará un aviso al usuario diciéndole que rellene todos los datos.
- 2 [Si el usuario tiene creado su restaurante]
 - 2.1 Se muestra la pantalla de información del restaurante con todos los datos del restaurante y de las mesas que están asociadas a el. En cada mesa tiene los botones de “Editar Mesa” y “Ver Código QR”.

- 2.2 [Si el usuario pulsa el botón “Editar Datos”]
 - 2.2.1 Se cargará la página de edición del restaurante con los datos del restaurante en los campos del formulario (Figura Restaurante 2).
 - 2.2.2 [Si el usuario pulsa el botón “Guardar Datos”]
 - 2.2.2.1 [Si todos los datos del formulario han sido rellenados]
 - 2.2.2.1.1 Se procede al guardado de los datos.
 - 2.2.2.1.2 Se mostrará la pantalla de información del restaurante con todos los datos (Figura Restaurante 1).
 - 2.2.2.2 [Si todos los datos del formulario no han sido rellenados]
 - 2.2.2.2.1 Se mostrará un aviso al usuario diciéndole que rellene todos los datos.
- 2.3 [Si el usuario pulsa el botón “Editar Mesa”]
 - 2.3.1 Se mostrará una ventana emergente que solicitará al usuario el número de personas de la mesa (Figura Restaurante 3).
 - 2.3.2 [Si el usuario pulsa “Aceptar”]
 - 2.3.2.1 [Si el usuario ha introducido un valor válido]
 - 2.3.2.1.1 Se muestra la pantalla de gestión de restaurantes con los datos de la mesa editados (Figura Restaurante 1).
 - 2.3.2.2 [Si el usuario no ha introducido un valor válido]
 - 2.3.2.2.1 Se muestra la pantalla de gestión de restaurantes sin realizar ningún cambio en la mesa (Figura Restaurante 1).
 - 2.3.3 [Si el usuario pulsa “Cancelar”]
 - 2.3.3.1 Se muestra la pantalla de gestión de restaurantes sin realizar ningún cambio en la mesa (Figura Restaurante 1).
- 2.4 [Si el usuario pulsa el botón “Ver Código QR”]
 - 2.4.1 Se mostrará la imagen QR correspondiente en una ventana emergente.
- 2.5 [Si el usuario pulsa el botón “Añadir Mesa”]
 - 2.5.1 Se mostrará una ventana emergente que solicitara al usuario el número de personas de la mesa (Figura Restaurante 3).
 - 2.5.2 [Si el usuario pulsa “Aceptar”]
 - 2.5.2.1 [Si el usuario ha introducido un valor válido]
 - 2.5.2.1.1 Se muestra la pantalla de gestión de restaurantes con los datos de la nueva mesa añadidos (Figura Restaurante 1).
 - 2.5.2.2 [Si el usuario no ha introducido un valor válido]
 - 2.5.2.2.1 Se muestra la pantalla de gestión de restaurantes sin añadir ninguna mesa (Figura Restaurante 1).
 - 2.5.3 [Si el usuario pulsa “Cancelar”]
 - 2.5.3.1 Se muestra la pantalla de gestión de restaurantes sin añadir ninguna mesa (Figura Restaurante 1).

Postcondiciones

Los datos creados/modificados/eliminados del restaurante han sido almacenados en el sistema.

Imágenes

61. Figura: Restaurante 1

Restaurante	Carta	Menú	Pedidos		Cerrar Sesión
-------------	-------	------	---------	--	---------------



Restaurante XXXXXXXXX
Descripción: XXXXXXXXXXXXXXXX

Editar Datos

Mesas

Mesa 0	4 Personas	Editar Mesa	Ver Código QR
Mesa 1	6 Personas	Editar Mesa	Ver Código QR

Añadir Mesa

62. Figura: Restaurante 2

Restaurante	Carta	Menú	Pedidos		Cerrar Sesión
-------------	-------	------	---------	--	---------------

Nombre:

Descripción:

Logo
 Seleccionar imagen

Guardar Datos

63. Figura: Restaurante 3



Caso de Uso Extendido Web: Gestionar Carta

64. Figura: CDU Extendido Web: Gestionar Carta



Descripción

El Usuario Identificado después de haber pulsado el botón “Carta” del menú superior tendrá acceso a la gestión de la carta del restaurante.

Actores

Usuario Web Identificado

Precondiciones

El usuario deberá tener creado un restaurante y que se haya pulsado en el botón Carta del menú superior.

Flujo de eventos

1. Se muestra una ventana con la lista de todos los elementos disponibles en la carta del restaurante separados por el tipo de plato. En cada uno de ellos están los botones “Editar” y “Eliminar”. En la parte superior hay un botón de “Añadir Artículo”.(Figura: Carta 1).
2. [Si el usuario pulsa el botón “Añadir Artículo”]
 - 2.1 Se cargará la página de edición del artículo con sus campos completamente vacíos (Figura: Carta 2).
 - 2.2 [Si el usuario pulsa el botón “Guardar Datos”]
 - 2.2.1 [Si todos los datos del formulario han sido rellenados]
 - 2.2.1.1 Se procede al guardado de los datos.
 - 2.2.1.2 Se mostrará la pantalla de la lista de artículos de la carta con el artículo añadido (Figura: Carta 1).
 - 2.2.2 [Si todos los datos del formulario no han sido rellenados]
 - 2.2.2.1 Se mostrará un aviso al usuario diciéndole que rellene todos los datos.
3. [Si el usuario pulsa el botón “Editar” en alguno de los artículos de la carta]
 - 3.1 Se cargará la página de edición del artículo con los datos del artículo a editar en los campos del formulario (Figura: Carta 2).

- 3.2 [Si el usuario pulsa el botón “Guardar Datos”]
 - 3.2.1 [Si todos los datos del formulario han sido rellenados]
 - 3.2.1.1 Se procede al guardado de los datos. Se mostrará la pantalla de la lista de artículos de la carta con el artículo editado (Figura: Carta 1).
 - 3.2.2 [Si todos los datos del formulario no han sido rellenados]
 - 3.2.2.1 Se mostrará un aviso al usuario diciéndole que rellene todos los datos.

- 4 [Si el usuario pulsa el botón “Eliminar” en alguno de los artículos de la carta]
 - 4.1 Se eliminará el artículo de la lista y se recargará la página de los artículos disponibles de la carta sin ese artículo.

Postcondiciones

Los datos creados/modificados/eliminados de los artículos del restaurante han sido almacenados en el sistema.

Imágenes

65. Figura: Carta 1



66. Figura: Carta 2

Restaurante	Carta	Menú	Pedidos		Cerrar Sesión
-------------	-------	------	---------	--	---------------

Nombre Artículo:	<input type="text" value="xxxxxxxx"/>
Descripción Artículo:	<input type="text" value="xxxxxxxx"/>
Precio Artículo:	<input type="text" value="xxxxx €"/>
Tipo de Artículo	<input type="text" value="xxxxxxxx"/> ▼
Imagen	<input type="text" value="xxxxxxxx.jpg"/> <input type="button" value="Seleccionar Imagen"/>
<input type="button" value="Guardar Datos"/>	

Caso de Uso Extendido Web: Gestionar Menú

67. Figura: CDU Extendido Web: Gestionar Menú



Descripción

El usuario identificado después de haber pulsado el botón “Menú” del menú superior tendrá acceso a la gestión de los menús del restaurante.

Actores

Usuario Web Identificado

Precondiciones

El usuario deberá tener creado un restaurante y que se haya pulsado en el botón Menú del menú superior.

Flujo de eventos

1. Se muestra una ventana con la lista de todos los menús disponibles en el restaurante. En cada uno de ellos están los botones “Editar” y “Eliminar”. En la parte superior hay un botón de “Añadir Menú” (Figura: Menú 1).
2. [Si el usuario pulsa el botón “Añadir Menú”]
 - 2.1 Se cargará la página de edición de menús con sus campos completamente vacíos (Figura: Menú 2).
 - 2.2 [Si el usuario pulsa el botón “Guardar Datos”]
 - 2.2.1 [Si todos los datos del formulario han sido rellenados]
 - 2.2.1.1 Se procede al guardado de los datos.
 - 2.2.1.2 Se mostrará la pantalla de la lista de menús del restaurante con el menú añadido (Figura: Menú 1).
 - 2.2.2 [Si todos los datos del formulario no han sido rellenados]
 - 2.2.2.1 Se mostrará un aviso al usuario diciéndole que rellene todos los datos.
3. [Si el usuario pulsa el botón “Editar” en alguno de los menús]
 - 3.1 Se cargará la página de edición de menús con los datos del menú a editar en los campos del formulario (Figura: Menú 2).

- 3.2 [Si el usuario pulsa el botón “Guardar Datos”]
 - 3.2.1 [Si todos los datos del formulario han sido rellenados]
 - 3.2.1.1 Se procede al guardado de los datos. Se mostrará la pantalla de la lista de menús del restaurante con el menú editado (Figura: Menú 1).
 - 3.2.2 [Si todos los datos del formulario no han sido rellenados]
 - 3.2.2.1 Se mostrará un aviso al usuario diciéndole que rellene todos los datos.
- 4. [Si el usuario pulsa el botón “Eliminar” en alguno de los menús]
 - 4.1 Se eliminara el menú de la lista y se recargará la página de los menús disponibles en el restaurante sin ese menú.

Postcondiciones

Los datos creados/modificados/eliminados de los menús del restaurante han sido almacenados en el sistema.

Imágenes

68. Figura: Menú 1



69. Figura: Menú 2

Restaurante	Carta	Menú	Pedidos		Cerrar Sesión
-------------	-------	-------------	---------	--	---------------

Nombre Menú:

Descripción Menú:

Precio Menú:

Artículos del Menú

Primeros

Primero X	Descripción XXXXX	XXXX €	<input checked="" type="checkbox"/>
Primero Y	Descripción YYYYY	XXXX €	<input type="checkbox"/>

Segundos

Segundo X	Descripción XXXXX	XXXX €	<input type="checkbox"/>
Segundo Y	Descripción YYYYY	XXXX €	<input checked="" type="checkbox"/>

Caso de Uso Extendido Web: Atender Pedido

70. Figura: CDU Extendido Web: Atender Pedido



Descripción

El usuario identificado después de haber pulsado el botón “Pedidos” del menú superior tendrá acceso a la gestión de los pedidos del restaurante.

Actores

Usuario Web Identificado

Precondiciones

El usuario deberá tener pedidos pendientes de atender, tener creado un restaurante y que se haya pulsado en el botón Pedidos del menú superior.

Flujo de eventos

- 1 Se cargará la página de gestión de pedidos en la cual se podrán ver los pedidos pendientes de atención y el botón “Atender” en cada uno de ellos (Figura Pedidos 1).
- 2 [Si el usuario pulsa en el botón en “Atender” de un pedido]
 - 2.1 Se mostrará una ventana emergente en la cual el usuario podrá especificar el mensaje que el usuario que ha realizado el pedido recibirá en su dispositivo (Figura Pedidos 2).
 - 2.2 [Si el usuario pulsa en “Enviar Mensaje”]
 - 2.2.1 Se enviará la notificación al usuario del pedido y se cargará la página de gestión de pedidos con el pedido atendido ya eliminado (Figura Pedidos 1).

Postcondiciones

Se habrá guardado en el sistema los pedidos que han sido atendidos por el usuario.

Imágenes

71. Figura: Pedidos 1

Restaurante	Carta	Menú	Pedidos	Cerrar Sesión
-------------	-------	------	---------	---------------

Pendientes de Atender

Pedido Z

Artículo Z	Descripción Z
Menú Z	Plato 12322 Bebida 12322 Postre 12322 Bebida 12322

Pendientes de Cobro

Pedido X	Usuario X	xx.xx €	<input type="button" value="Cobrar"/>
Pedido Y	Usuario Y	xx.xx €	<input type="button" value="Cobrar"/>

72. Figura: Pedidos 2

Restaurante	Carta	Menú	Pedidos	Cerrar Sesión
-------------	-------	------	---------	---------------

Pendientes de Atender

Pedido Z

Artículo Z	Descripción Z
Menú Z	Plato 12322 Bebida 12322 Postre 12322 Bebida 12322

Pendientes de Cobro

Pedido X	Usuario X	xx.xx €	<input type="button" value="Cobrar"/>
Pedido Y	Usuario Y	xx.xx €	<input type="button" value="Cobrar"/>

Inserta el mensaje que recibirá el usuario en forma de notificación.

Su pedido ha sido atendido. En breves minutos lo tendrá en la mesa

Caso de Uso Extendido Web: Realizar Cobro

73. Figura: CDU Extendido Web: Realizar Cobro



Descripción

El usuario identificado después de haber pulsado el botón “Pedidos” del menú superior tendrá acceso a la gestión de los Pedidos del restaurante.

Actores

Usuario Web Identificado

Precondiciones

El usuario deberá tener pedidos pendientes de cobrar, tener creado un restaurante y que se haya pulsado en el botón Pedidos del menú superior.

Flujo de eventos

- 1 Se cargará la página de gestión de pedidos en la cual se podrán ver los pedidos pendientes de cobrar y el botón “Cobrar” en cada uno de ellos (Figura Pedidos 1).
- 2 [Si el usuario pulsa en el botón en “Cobrar” de un pedido]
 - 2.1 Se notificará al usuario del pedido y se cargará de nuevo la página de gestión de pedidos con el pedido cobrado ya eliminado (Figura Pedidos 1).

Postcondiciones

Se habrá guardado en el sistema los pedidos que han sido cobrados por el usuario.

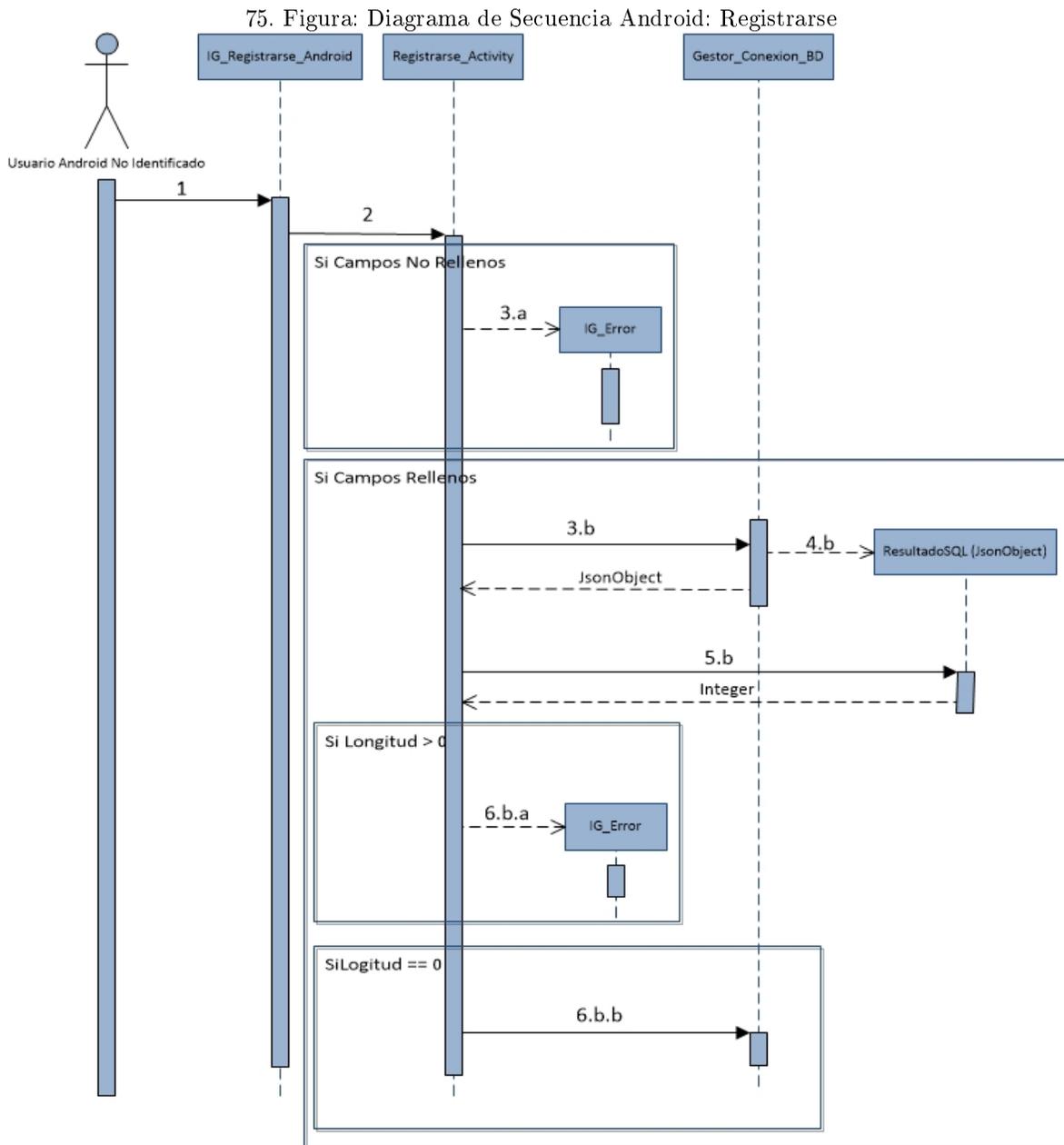
Imágenes

74. Figura: Pedidos 1

Restaurante	Carta	Menú	Pedidos	Cerrar Sesión
Pendientes de Atender				
Pedido Z Atender				
Artículo Z	Descripción Z			
	Pedido: 2222			
	Reporte: 2222			
Menú Z	Pedido: 2222			
	Detalle: 2222			
Pendientes de Cobro				
Pedido X	Usuario X	xx,xx €	Cobrar	
Pedido Y	Usuario Y	xx,xx €	Cobrar	

ANEXO II.- Diagramas de Secuencia

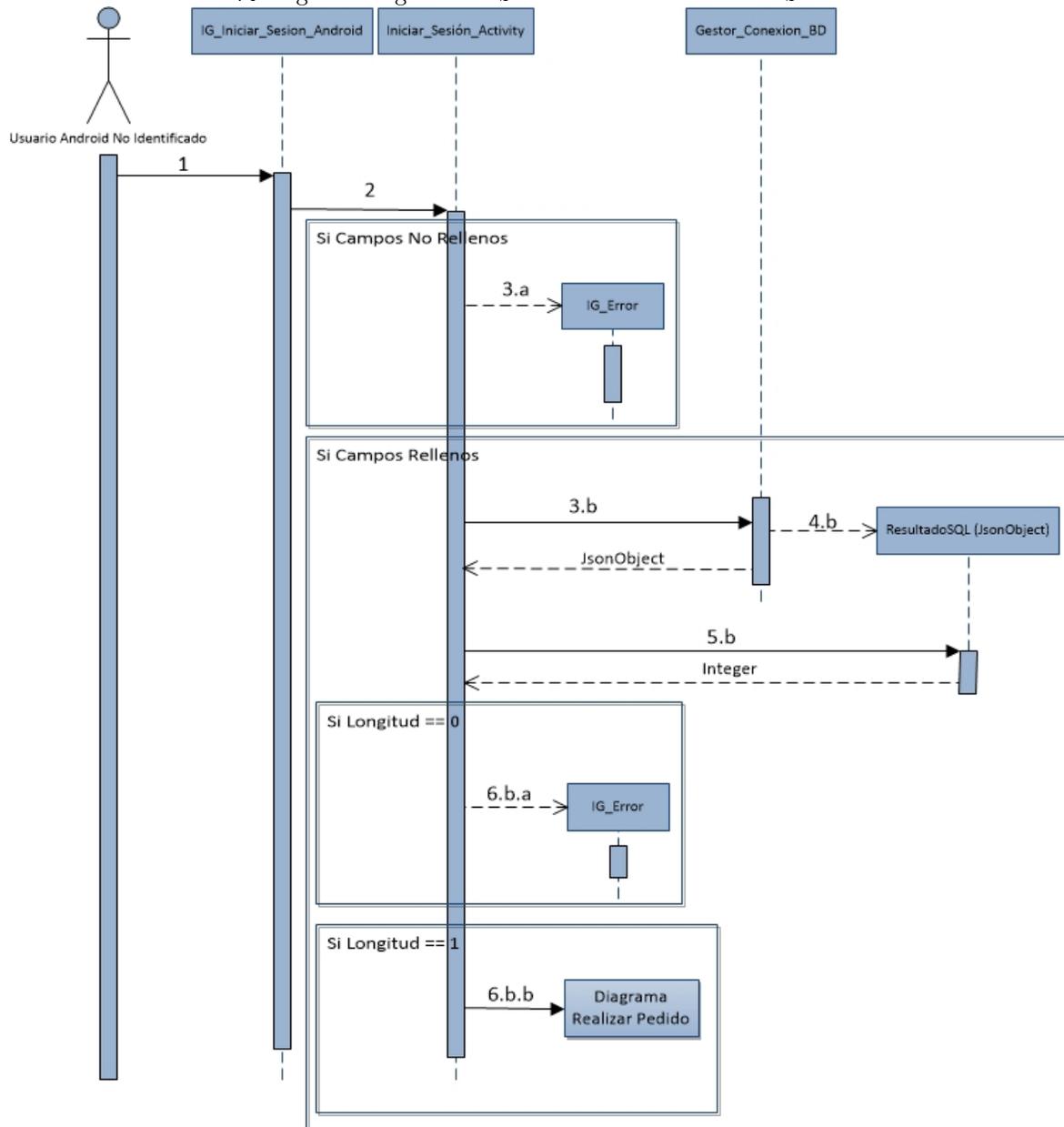
Diagrama de Secuencia Android: Registrarse



1: El usuario hace clic en Registrarse.
2: onButtonClicked();
[Si el usuario ha introducido todos los datos]
3.a: new("No se han introducido todos los datos requeridos");
[Si el usuario NO ha introducido todos los datos]
3.b: ObtenerUsuariosPorNombre(nombreUsuario) : JsonObject;
4.b: new();
5.b: length();
[Si el la longitud de la respuesta de la BD es Mayor de 0]
6.a: new("El usuario ya existe en la base de datos");
[Si el la longitud de la respuesta de la BD es Igual a 0]
6.b: CrearUsuario(datosUsuario);

Diagrama de Secuencia Android: Iniciar Sesión

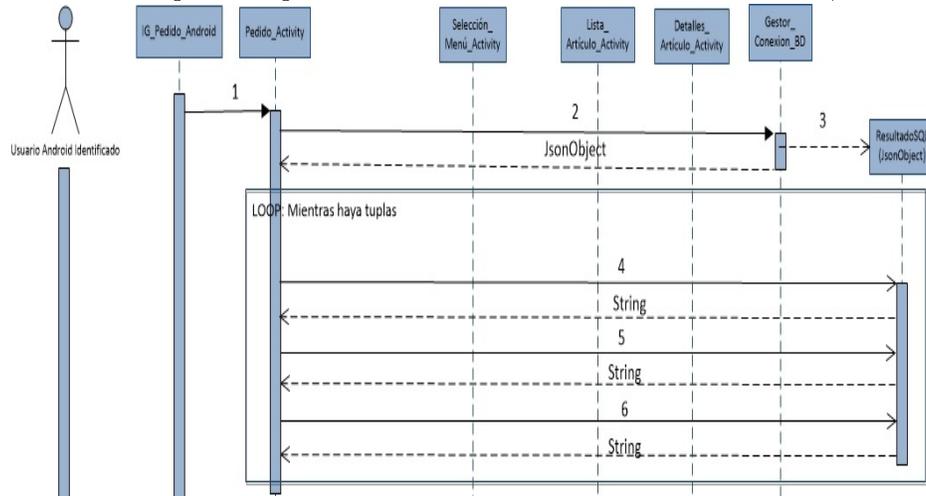
76. Figura: Diagrama de Secuencia Android: Iniciar Sesión



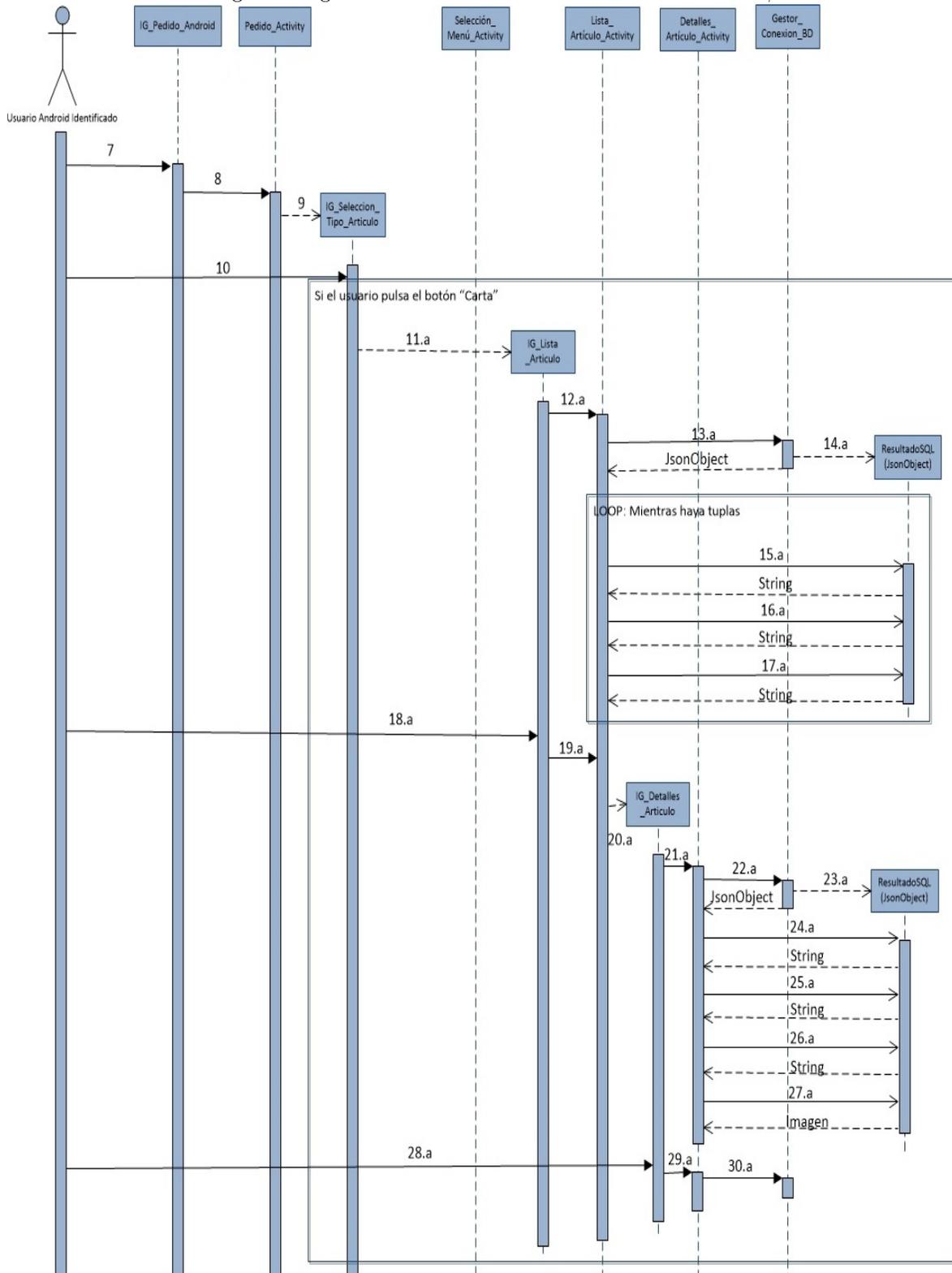
1: El usuario hace clic en Iniciar Sesión.
 2: `onButtonClicked()`;
 [Si el usuario ha introducido todos los datos]
 3.a: `new("No se han introducido todos los datos requeridos")`;
 [Si el usuario NO ha introducido todos los datos]
 3.b: `ObtenerUsuario(nombreUsuario, passwordUsuario) : JsonObject`;
 4.b: `new()`;
 5.b: `length()`;
 [Si el la longitud de la respuesta de la BD es Igual a 0]
 6.b.a: `new("El usuario ya existe en la base de datos")`;
 [Si el la longitud de la respuesta de la BD es Igual a 1]
 6.b.b: Acciones del diagrama de Secuencia Realizar Pedido

Diagrama de Secuencia Android: Realizar Pedido

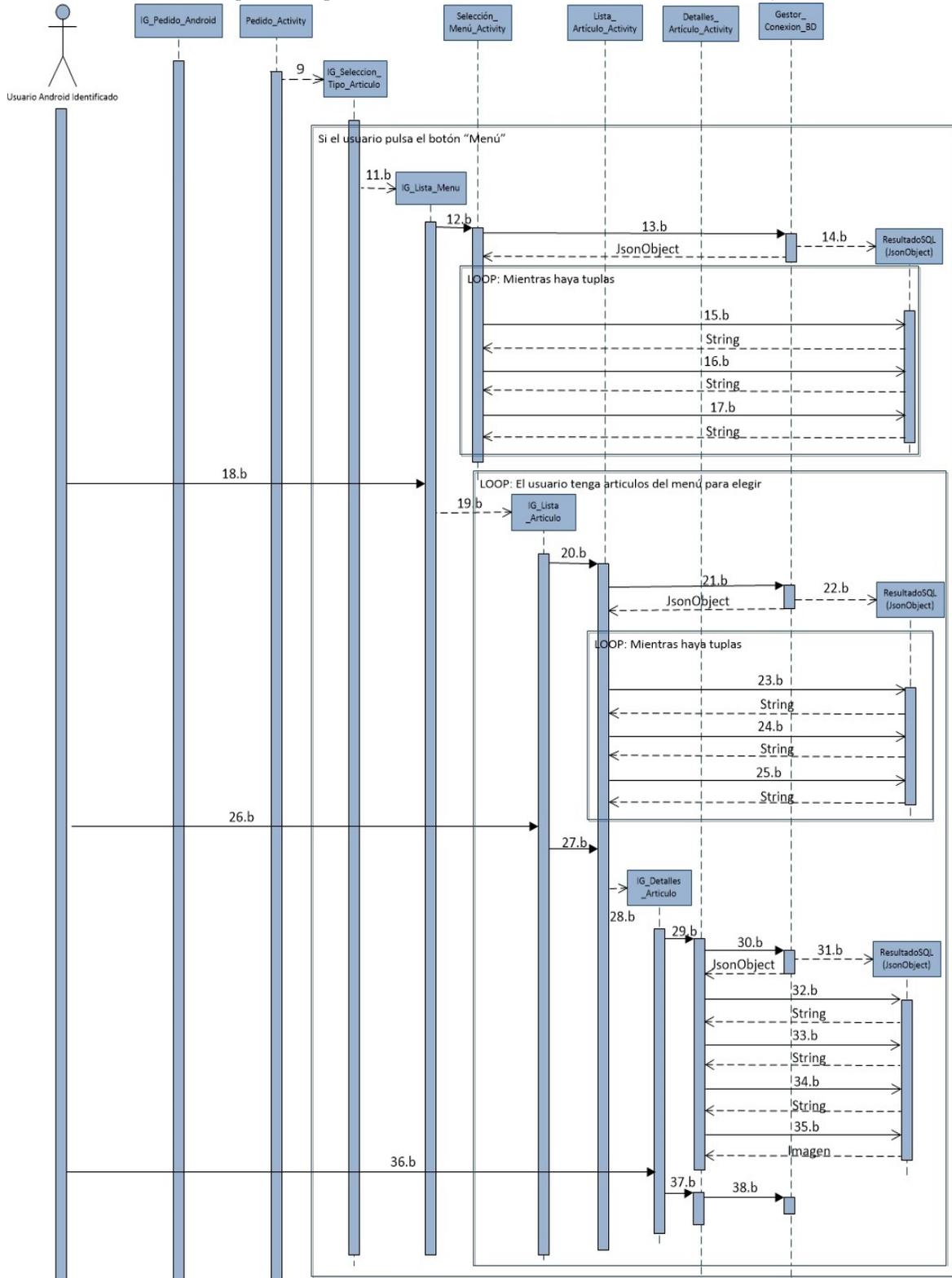
77. Figura: Diagrama de Secuencia Android: Realizar Pedido 1/6



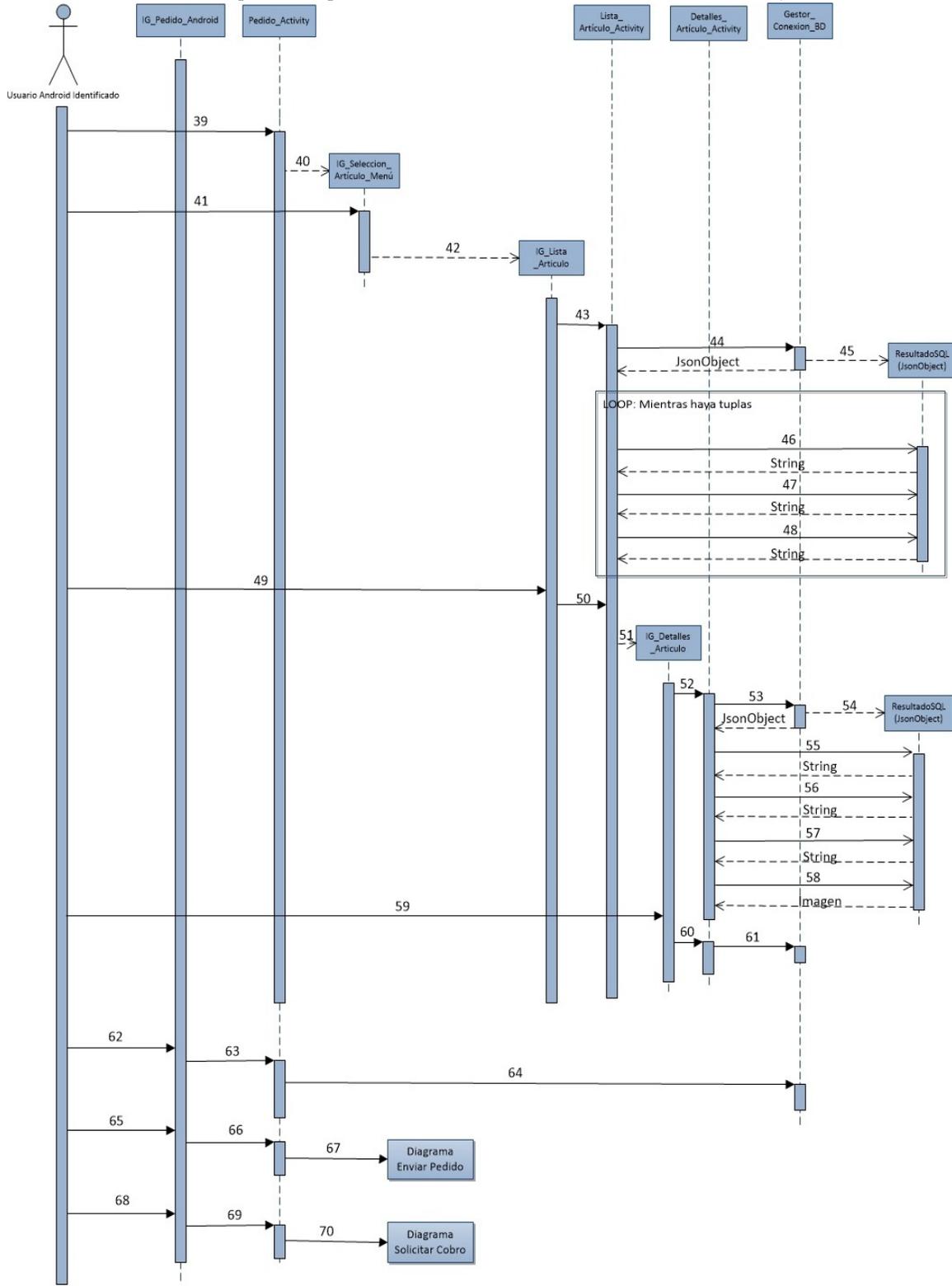
78. Figura: Diagrama de Secuencia Android: Realizar Pedido 2/6



79. Figura: Diagrama de Secuencia Android: Realizar Pedido 3/6



80. Figura: Diagrama de Secuencia Android: Realizar Pedido 4/6



81. Figura: Diagrama de Secuencia Android: Realizar Pedido 5/6

```

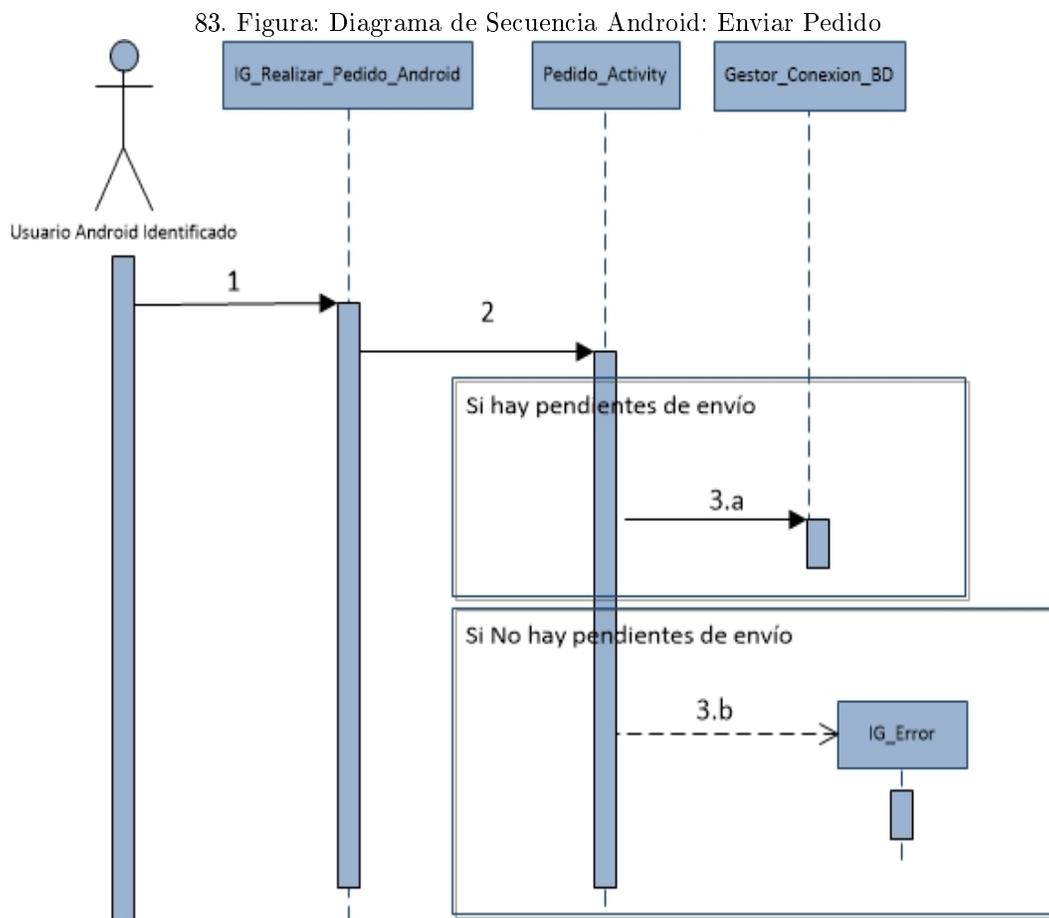
1: cargarDatosPedido();
2: obtenerArticulosPedido(idTicket) : JsonObject;
3: new();
[LOOP Mientras haya tuplas]
4: getString("nombreArticulo") : String;
5: getString("descripcionArticulo") : String;
6: getString("precioArticulo") : String;
7: El usuario pulsa en el botón "Añadir".
8: seleccionarTipoArticulo();
9: new();
10: El usuario selecciona una opción de la ventana emergente.
[Si el usuario pulsa el botón "Carta"]
11.a: new();
12.a: obtenerArticulosRestaurante();
13.a: obtenerArticulosRestauranteDeLaBD(idRestaurante) : JsonObject;
14.a: new();
[LOOP Mientras haya tuplas]
15.a: getString("nombreArticulo") : String;
16.a: getString("descripcionArticulo") : String;
17.a: getString("tipoArticulo") : String;
18.a: El usuario selecciona un articulo de la carta.
19.a: verDetalles(nombreArticulo);
20.a: new();
21.a: obtenerArticulo(nombreArticulo,idRestaurante);
22.a: obtenerArticuloDeLaBD(nombreArticulo,idRestaurante) : JsonObject;
23.a: new();
24.a: getString("nombreArticulo") : String;
25.a: getString("descripcionArticulo") : String;
26.a: getString("precio") : String;
27.a: getString("imagen") : Image;
28.a: El usuario pulsa el botón "Añadir".
29.a: añadirArticuloAlPedido(idArticulo,idPedido);
30.a: añadirArticuloAlPedidoEnLaBD(idArticulo,idPedido);
[Si el usuario pulsa el botón Menú]
11.b: new();
12.b: obtenerMenúsRestaurante();
13.b: obtenerMenúsRestauranteDeLaBD(idRestaurante) : JsonObject;
14.b: new();
[LOOP Mientras haya tuplas]
15.b: getString("nombreMenú") : String;
16.b: getString("descripcionMenú") : String;
17.b: getString("precio") : String;
18.b: El usuario selecciona uno de los Menús.
[LOOP Mientras el Usuario tenga artículos del menú para elegir]
19.b: new(idMenu,platoAElegir);
20.b: obtenerArticulosDelMenu(idMenú,platoAElegir);
21.b: obtenerArticulosDelMenuDeLaBD(idMenú,platoAElegir,idRestaurante) : JsonObject;
22.b: new();
[LOOP Mientras haya tuplas]
23.b: getString("nombreArticulo") : String;
24.b: getString("descripcionArticulo") : String;
25.b: getString("tipoArticulo") : String;
26.b: El usuario selecciona un articulo del menu.
27.b: verDetalles(nombreArticulo);
28.b: new();
29.b: obtenerArticulo(nombreArticulo,idRestaurante);
30.b: obtenerArticuloDeLaBD(nombreArticulo,idRestaurante) : JsonObject;
31.b: new();
32.b: getString("nombreArticulo") : String;
33.b: getString("descripcionArticulo") : String;
34.b: getString("precio") : String;
35.b: getString("imagen") : Image;
36.b: El usuario pulsa el botón "Añadir".
37.b: añadirArticuloAlMenu(idArticulo,idMenú,idPedido);
38.b: añadirArticuloAlPedidoEnLaBD(idArticulo,idMenú,idPedido);

```

82. Figura: Diagrama de Secuencia Android: Realizar Pedido 6/6

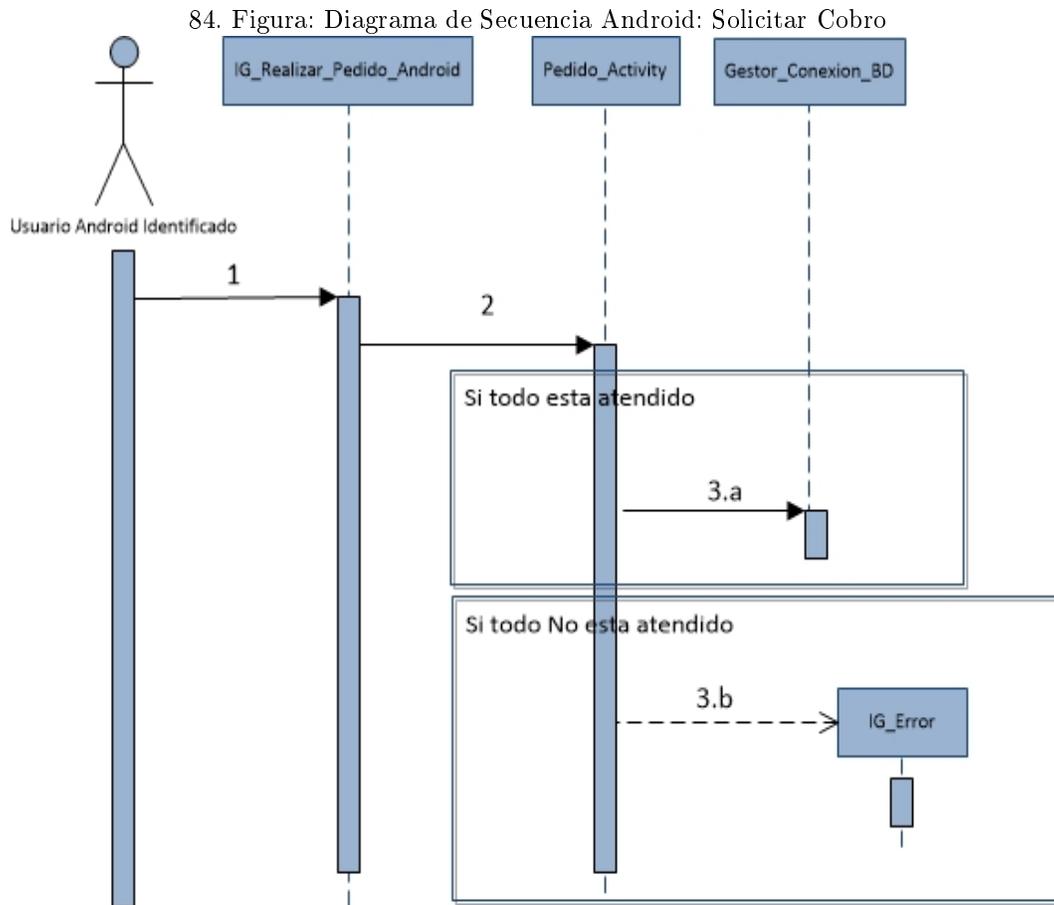
```
39: El usuario pulsa en "Editar" en alguno de los menús.
40: new();
41: El usuario selecciona el plato que desea modificar.
42: new(idMenú, tipoPlato)
43: obtenerArticulosDelMenu(idMenú, platoAElegir);
44: obtenerArticulosDelMenuDeLaBD(idMenú, platoAElegir, idRestaurante) : JsonObject;
45: new();
[LOOP Mientras haya tuplas]
 46: getString("nombreArticulo") : String;
 47: getString("descripcionArticulo") : String;
 48: getString("tipoArticulo") : String;
49: El usuario selecciona un articulo del menu.
50: verDetalles(nombreArticulo);
51: new();
52: obtenerArticulo(nombreArticulo, idRestaurante);
53: obtenerArticuloDeLaBD(nombreArticulo, idRestaurante) : JsonObject;
54: new();
55: getString("nombreArticulo") : String;
56: getString("descripcionArticulo") : String;
57: getString("precio") : String;
58: getString("imagen") : Image;
59: El usuario pulsa el botón "Añadir".
60: añadirArticuloAlMenu(idArticulo, idMenú, idPedido);
61: añadirArticuloAlPedidoEnLaBD(idArticulo, idMenú, idPedido);
62: El usuario pulsa en eliminar una articulo del pedido
63: eliminarArticuloOMenu(idArticuloOMenu);
64: eliminarArticuloOMenuDeLaBD(idArticuloOMenu);
65: El usuario pulsa en el botón "Enviar Pedido".
66: enviarPedido();
67: Acciones del diagrama de Secuencia Enviar Pedido.
68: El usuario pulsa en el botón "Solicitar Cobro".
69: solicitarCobro();
70: Acciones del diagrama de Secuencia Solicitar Cobro.
```

Diagrama de Secuencia Android: Enviar Pedido



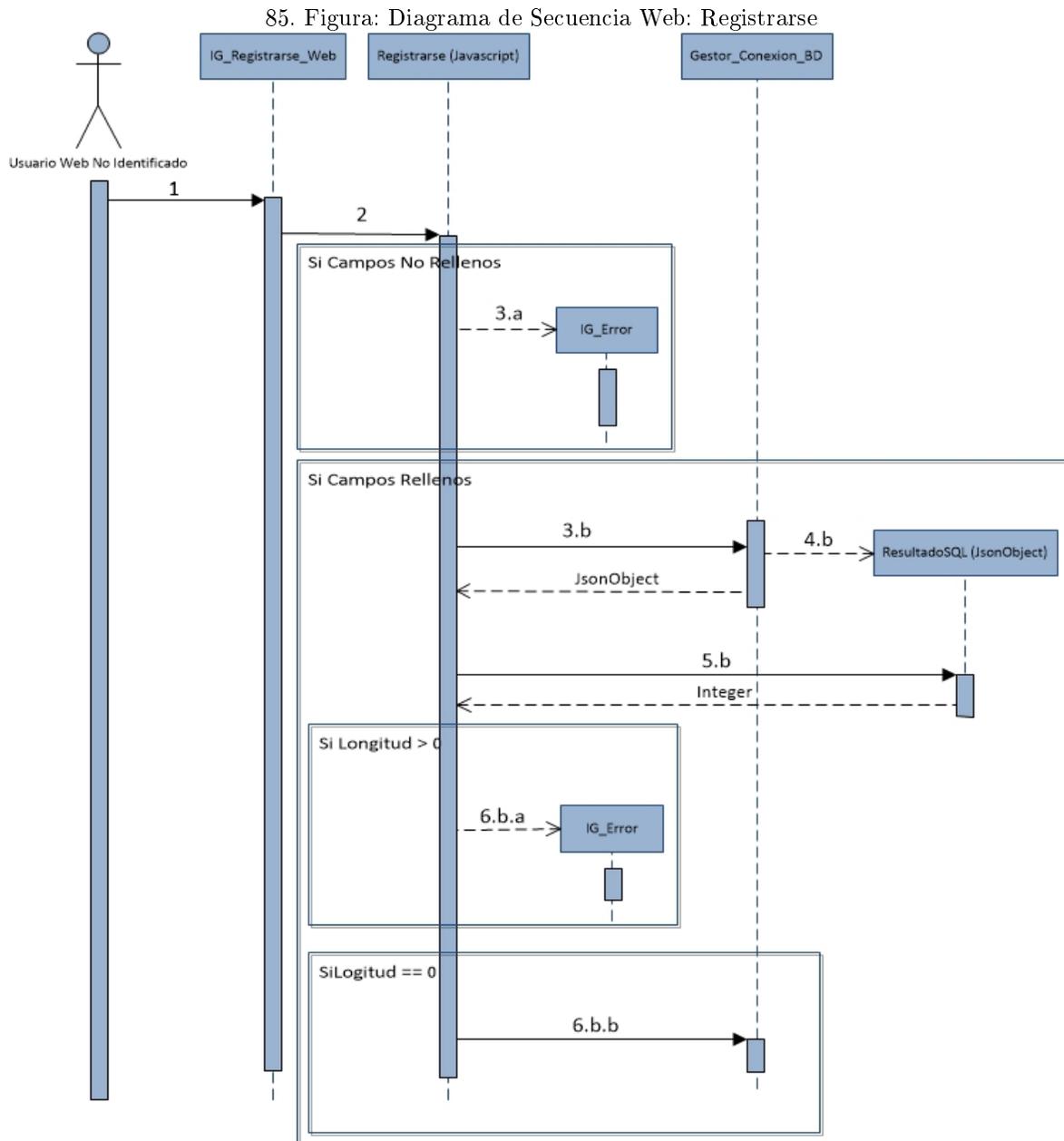
1: El usuario hace clic en Enviar Pedido.
2: `onButtonClicked()`;
[Si el pedido tiene artículos pendientes de envío]
3.a: `cambiarEstadoAEnviado(listaArticulosPendientes)`;
[Si el pedido No tiene artículos pendientes de envío]
3.b: `new("No se han introducidos todos los datos requeridos")`;

Diagrama de Secuencia Android: Solicitar Cobro



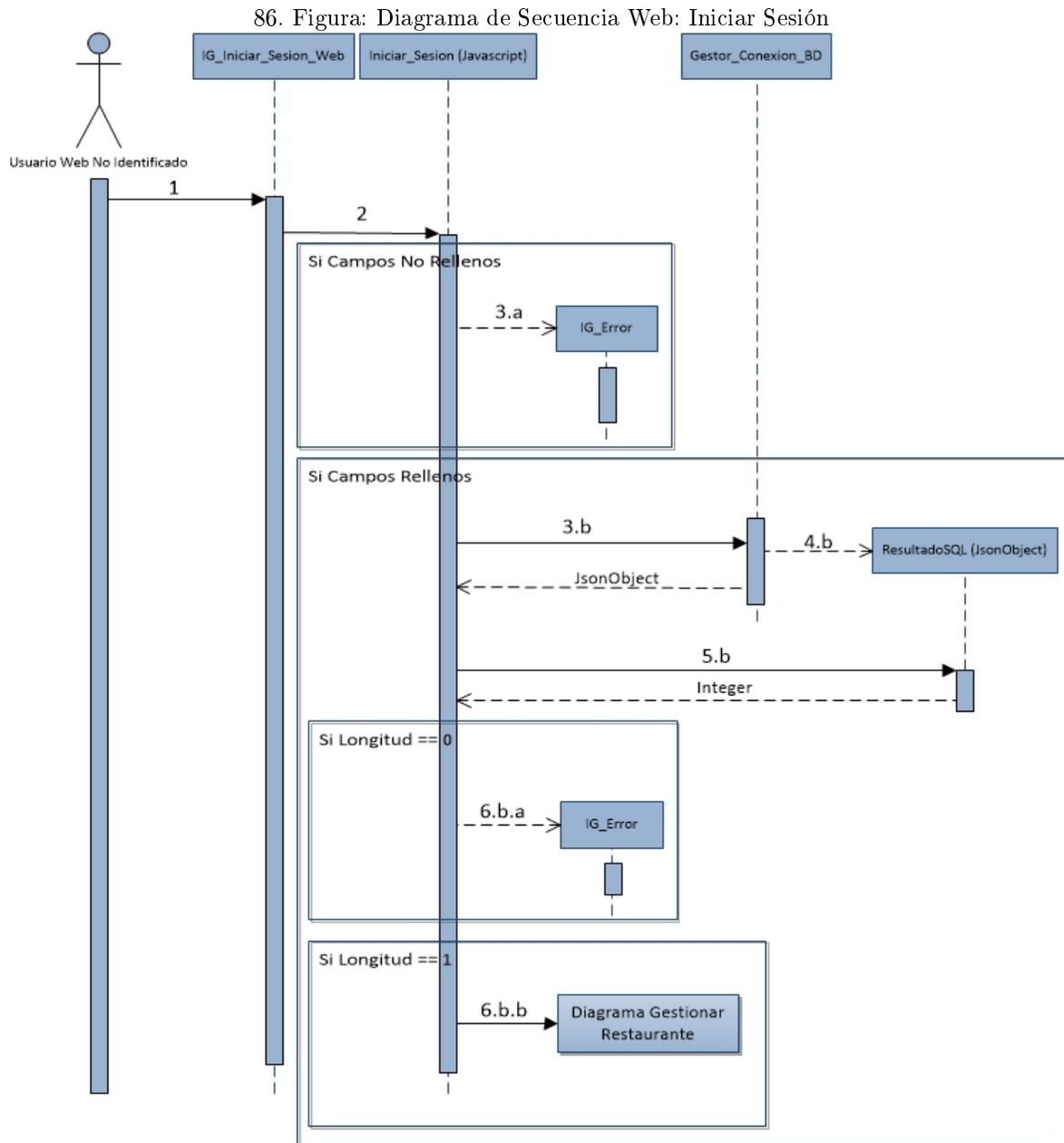
1: El usuario hace clic en Solicitar Cobro.
2: `onButtonClicked()`;
[Si los artículos del pedido están atendidos]
3.a: `solicitarCobroTicket(idTicket)`;
[Si los artículos del pedido No están atendidos]
3.b: `new("Todos los artículos deben estar atendidos para poder proceder con el cobro")`;

Diagrama de Secuencia Web: Registrarse



1: El usuario hace clic en Registrarse.
 2: registrarse();
 [Si el usuario ha introducido todos los datos]
 3.a: new("No se han introducido todos los datos requeridos");
 [Si el usuario NO ha introducido todos los datos]
 3.b: ObtenerUsuariosPorNombre(nombreUsuario) : JsonObject;
 4.b: new();
 5.b: length();
 [Si el la longitud de la respuesta de la BD es Mayor de 0]
 6.b.a: new("El usuario ya existe en la base de datos");
 [Si el la longitud de la respuesta de la BD es Igual a 0]
 6.b.b: CrearUsuario(datosUsuario);

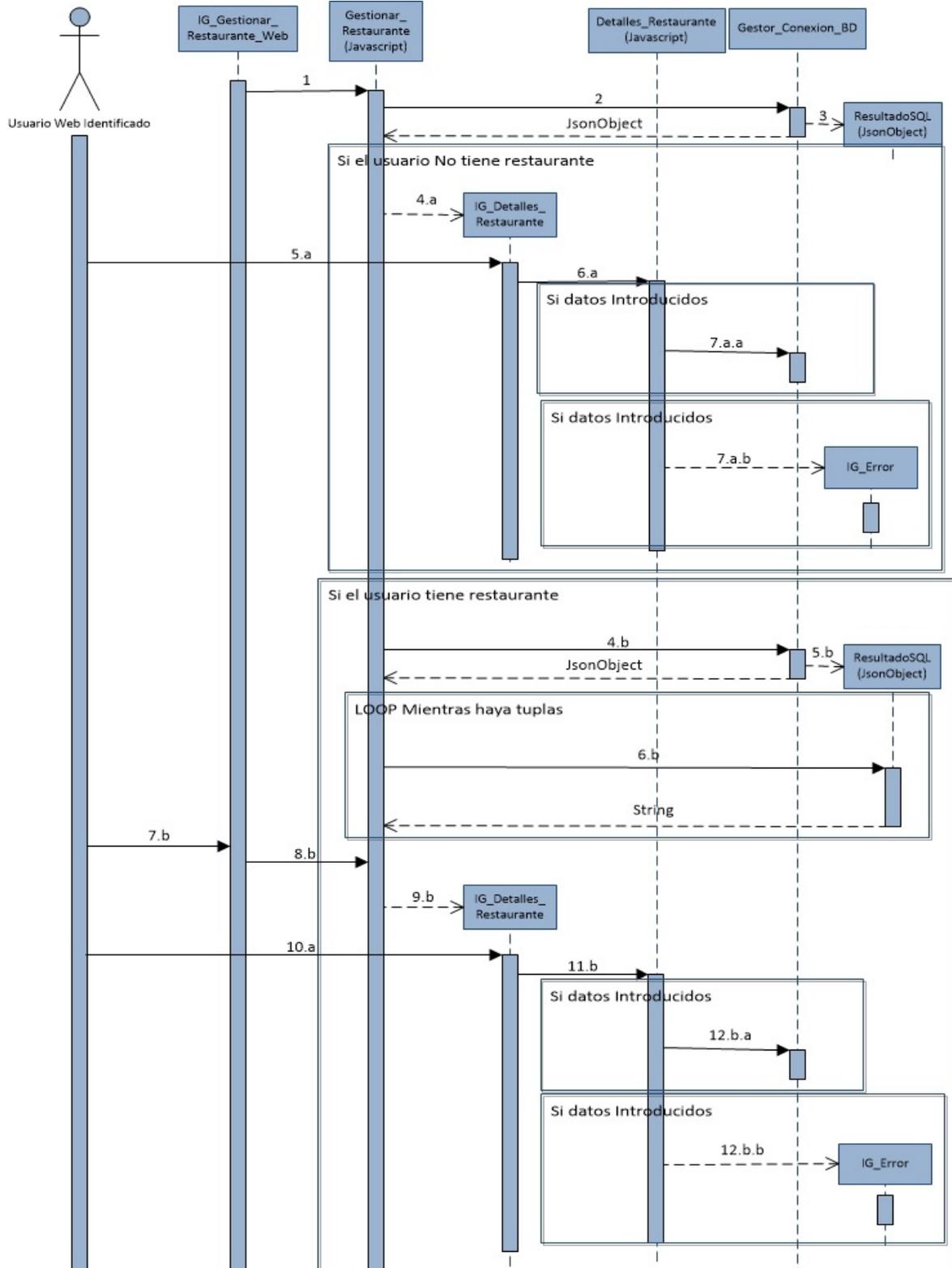
Diagrama de Secuencia Web: Iniciar Sesión



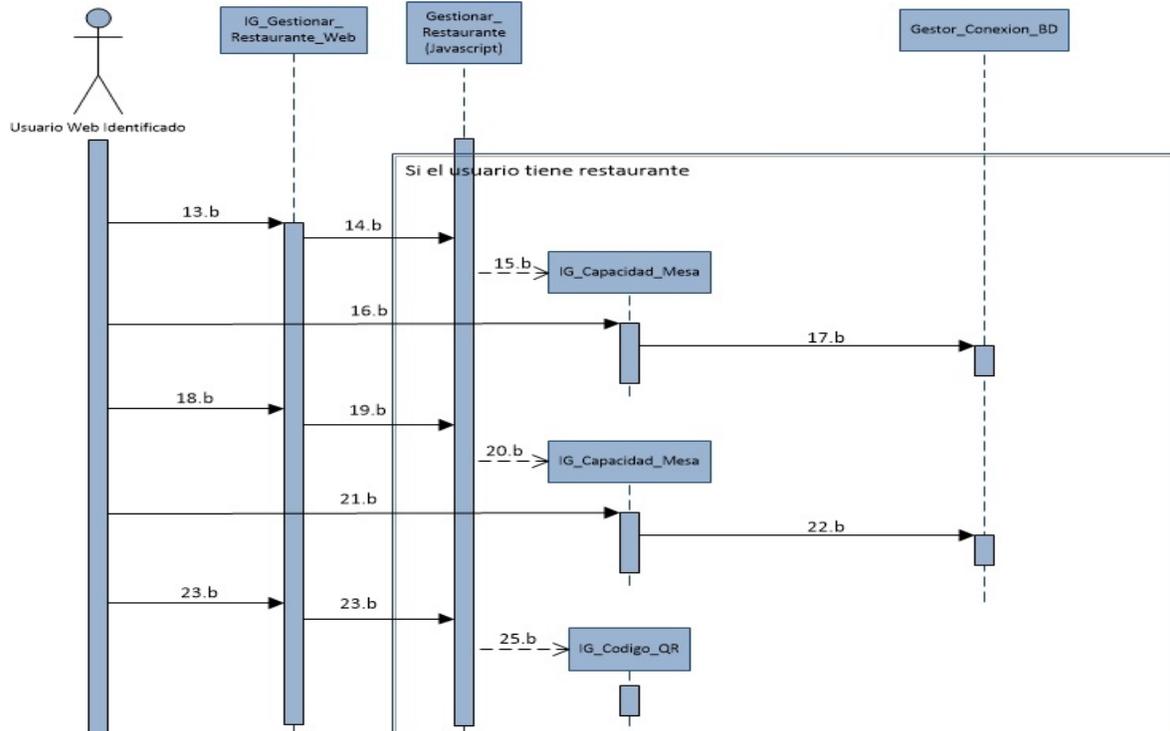
1: El usuario hace clic en Iniciar Sesión.
 2: `onButtonClicked()`;
 [Si el usuario ha introducido todos los datos]
 3.a: `new("No se han introducido todos los datos requeridos")`;
 [Si el usuario NO ha introducido todos los datos]
 3.b: `ObtenerUsuario(nombreUsuario, passwordUsuario) : JsonObject`;
 4.b: `new()`;
 5.b: `length()`;
 [Si el la longitud de la respuesta de la BD es Igual a 0]
 6.b.a: `new("El usuario ya existe en la base de datos")`;
 [Si el la longitud de la respuesta de la BD es Igual a 1]
 6.b.b: Acciones del diagrama de Secuencia Gestionar Restaurante.

Diagrama de Secuencia Web: Gestionar Restaurante

87. Figura: Diagrama de Secuencia Web: Gestionar Restaurante 1/2



88. Figura: Diagrama de Secuencia Web: Gestionar Restaurante 2/2

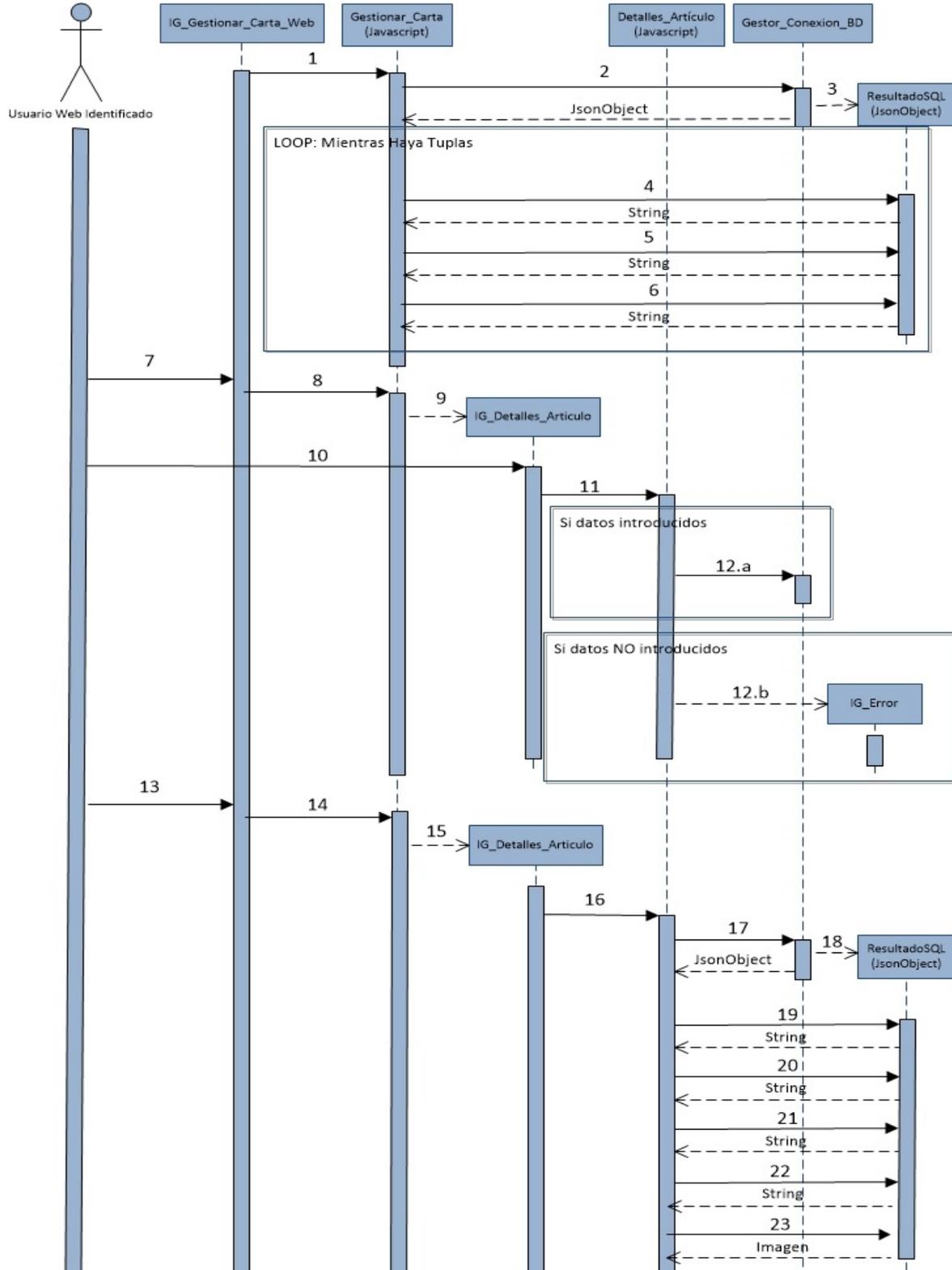


```

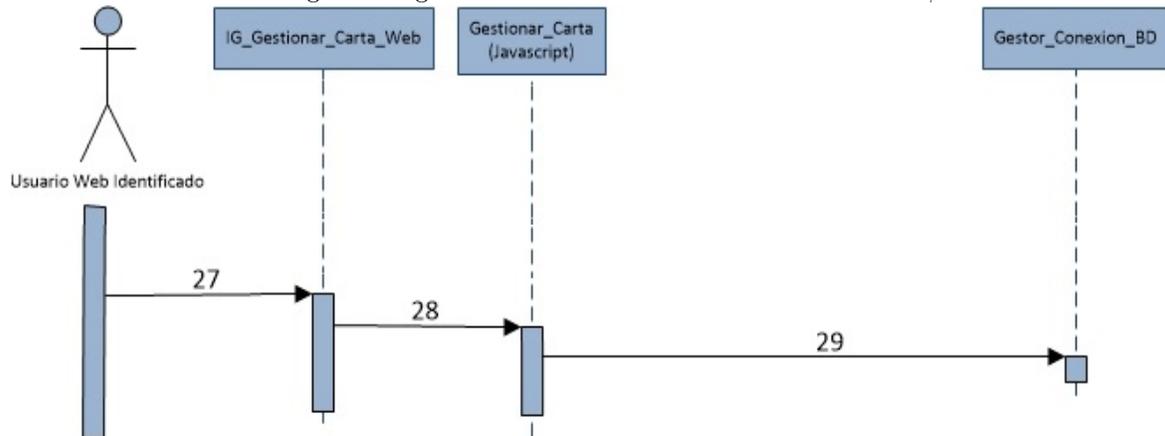
1: CargarDatosRestaurante();
2: obtenerRestauranteDelUsuario(idUsuario) : JsonObject;
3: new();
[Si el usuario No tiene restaurante]
4.a: new();
5.a: El usuario introduce los datos del restaurante y pulsa en "Guardar Datos"
6.a: añadirRestaurante(idRestaurante);
[Si el usuario ha introducido todos los datos]
7.a.a: añadirRestauranteALaBD(datosRestaurante);
[Si el usuario No ha introducido todos los datos]
7.a.b: new("No has introducido todos los datos");
[Si el usuario tiene restaurante]
4.b: obtenerMesasDelRestaurante(idRestaurante) : JsonObject;
5.b: new();
[LOOP Mientras haya tuplas]
6.b: getString("Capacidad") : String;
7.b: El usuario pulsa en "Editar Datos" del restaurante.
8.b: editarRestaurante(idRestaurante);
9.b: new();
10.b: El usuario introduce los datos del restaurante y pulsa en "Guardar Datos"
11.b: editarRestaurante();
[Si el usuario ha introducido todos los datos]
12.b.a: EditarRestauranteEnLaBD(datosRestaurante);
[Si el usuario No ha introducido todos los datos]
12.b.b: new("No has introducido todos los datos");
13.b: El usuario pulsa en "Añadir Mesa"
14.b: añadirMesaAlRestaurante();
15.b: new();
16.b: El usuario introduce la capacidad de la mesa y pulsa "Aceptar".
17.b: añadirMesaALaBD(idRestaurante,capacidad);
18.b: El usuario pulsa en "Editar Mesa"
19.b: editarMesaDelRestaurante(idMesa);
20.b: new();
21.b: El usuario introduce la capacidad de la mesa y pulsa "Aceptar".
22.b: editarMesaALaBD(idMesa,capacidad);
23.b: El usuario pulsa en el botón de "Ver Codigo QR" de una mesa.
24.b: mostrarCodigoQR(idMesa);
25.b: new();
    
```

Diagrama de Secuencia Web: Gestionar Carta

89. Figura: Diagrama de Secuencia Web: Gestionar Carta 1/2



90. Figura: Diagrama de Secuencia Web: Gestionar Carta 2/2

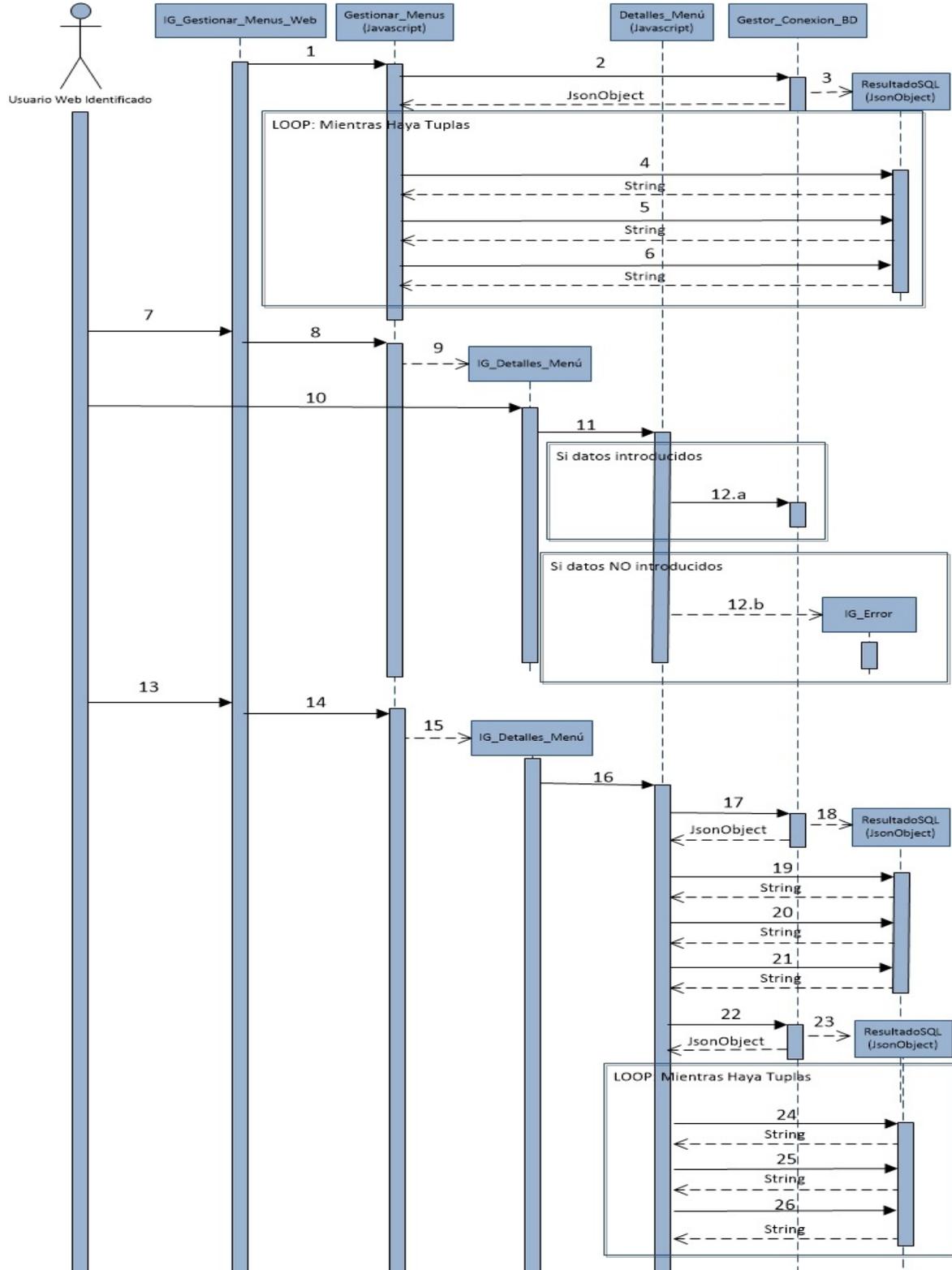


```

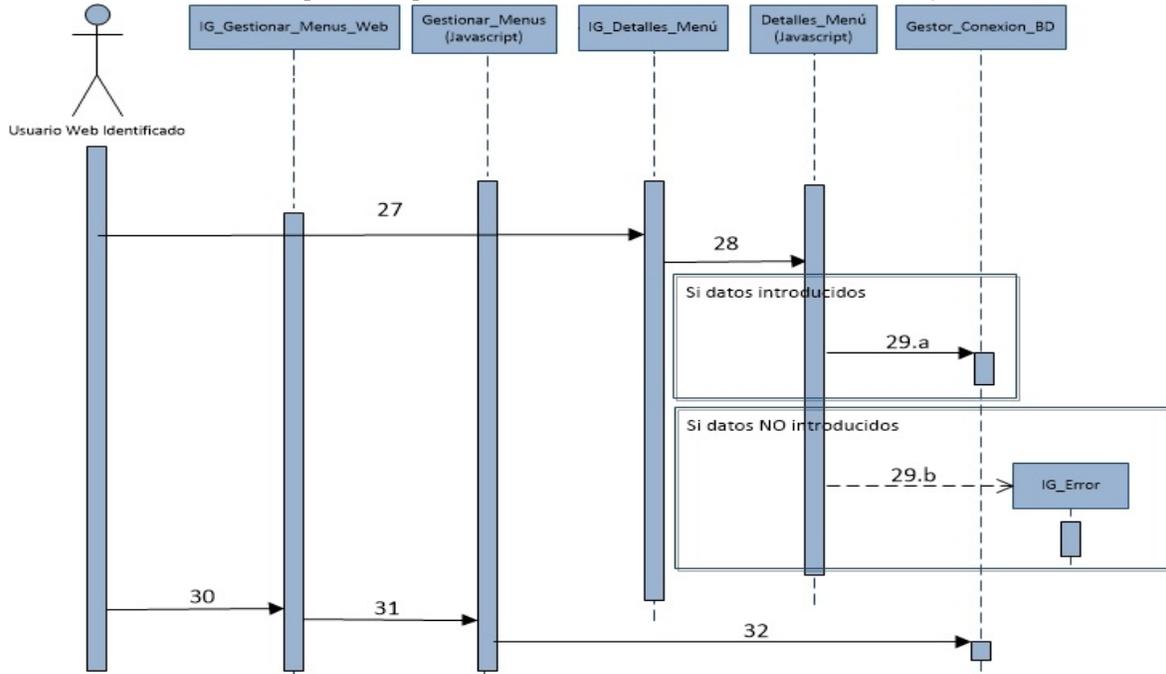
1: cargarArticulos(IdRestaurante);
2: obtenerArticulosDelRestaurante(idRestaurante) : JsonObject;
3: new();
[LOOP Mientras haya tuplas]
4: getString("nombreArticulo") : String;
5: getString("descripcionArticulo") : String;
6: getString("precio") : String;
7: El usuario pulsa el botón "Añadir Artículo".
8: añadirArticulo();
9: new();
10: El usuario Introduce los datos y pulsa en el botón "Guardar Datos".
11: guardarDatos();
[Si Todos los datos han sido introducidos]
12.a: añadirArticuloALaBD(datosArticulo);
[Si Todos los datos han No sido introducidos]
12.b: new();
13: El usuario pulsa el botón "Editar" de un artículo.
14: editarArticulo(idArticulo);
15: new();
16: cargarDatosArticulo(idArticulo);
17: obtenerArticulo(idArticulo)
18: new();
19: getString("nombreArticulo") : String;
20: getString("descripcionArticulo") : String;
21: getString("precio") : String;
22: getString("tipoArticulo") : String;
23: getString("imagen") : String;
24: El usuario Introduce los datos y pulsa en el botón "Guardar Datos".
25: guardarDatos();
[Si Todos los datos han sido introducidos]
26.a: editarArticuloDeLaBD(datosArticulo);
[Si Todos los datos No han sido introducidos]
26.b: new();
27: El usuario pulsa el botón "Eliminar" de un artículo.
28: eliminarArticulo(idArticulo);
29: eliminarArticuloDeLaBD(idArticulo);
    
```

Diagrama de Secuencia Web: Gestionar Menús

91. Figura: Diagrama de Secuencia Web: Gestionar Menús 1/2



92. Figura: Diagrama de Secuencia Web: Gestionar Menús 2/2

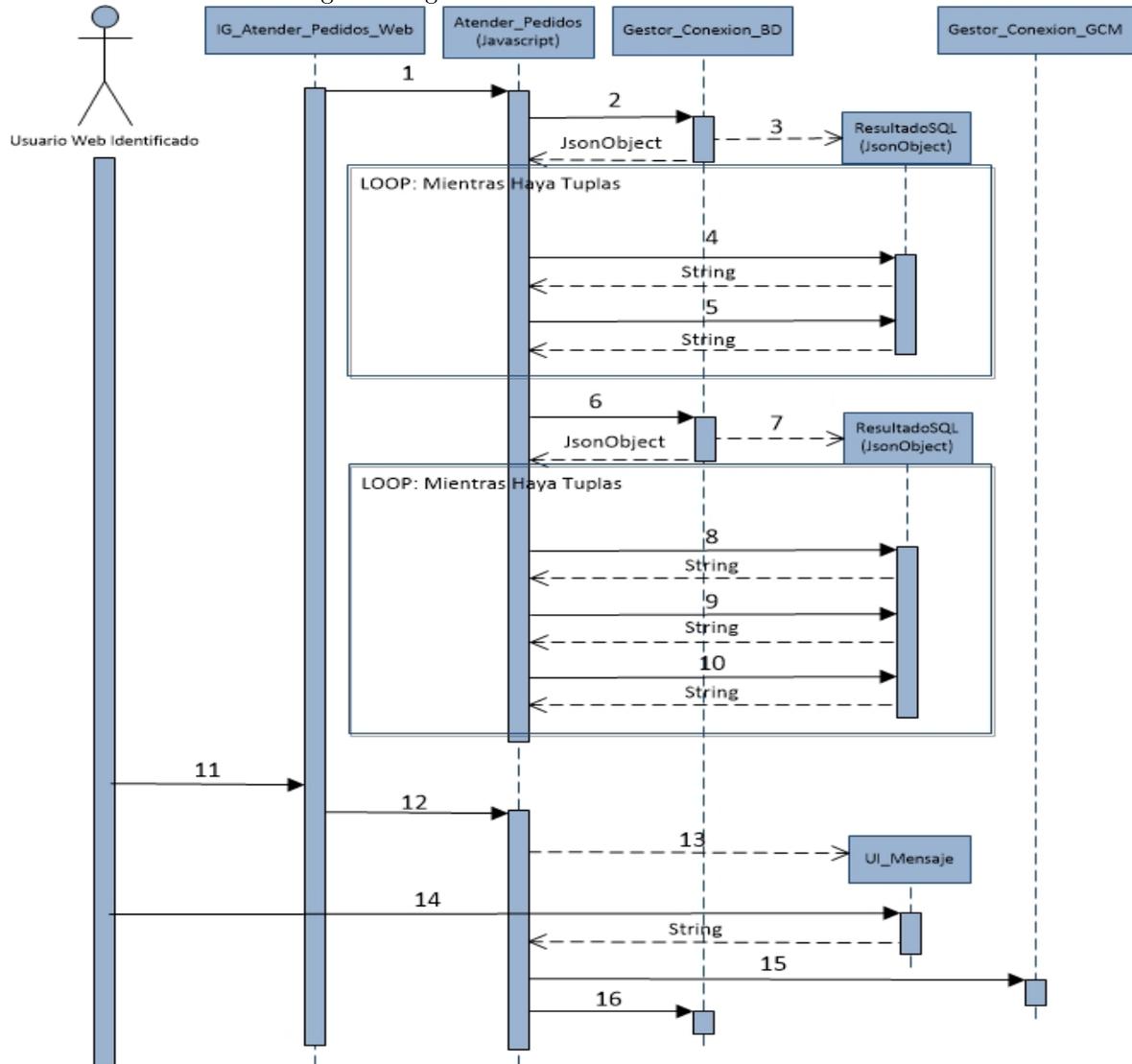


```

1: cargarMenús(idRestaurante);
2: obtenerMenúsDelRestaurante(idRestaurante) : JsonObject;
3: new();
[LOOP Mientras haya tuplas]
4: getString("nombreMenú") : String;
5: getString("descripcionMenú") : String;
6: getString("precio") : String;
7: El usuario pulsa el botón "Añadir Menú".
8: añadirMenú();
9: new();
10: El usuario Introduce los datos y pulsa en el botón "Guardar Datos".
11: guardarDatos();
[Si Todos los datos han sido introducidos]
12.a: añadirMenúAlaBD(datosMenú);
[Si Todos los datos No han sido introducidos]
12.b: new();
13: El usuario pulsa el botón "Editar" de un menú.
14: editarMenú(idMenú);
15: new();
16: cargarDatosMenú(idMenú);
17: obtenerMenú(idMenú)
18: new();
19: getString("nombreMenú") : String;
20: getString("descripcionMenú") : String;
21: getString("precio") : String;
22: obtenerArticulosDelMenú(idMenú)
23: new();
[LOOP Mientras haya tuplas]
24: getString("nombreArticulo") : String;
25: getString("descripcionArticulo") : String;
26: getString("precio") : String;
27: El usuario Introduce los datos y pulsa en el botón "Guardar Datos".
28: guardarDatos();
[Si Todos los datos han sido introducidos]
29.a: editarMenúDeLaBD(datosMenú);
[Si Todos los datos No han sido introducidos]
29.b: new();
30: El usuario pulsa el botón "Eliminar" de un menú.
31: eliminarMenú(idMenú);
32: eliminarMenúDeLaBD(idMenú);
    
```

Diagrama de Secuencia Web: Atender Pedido

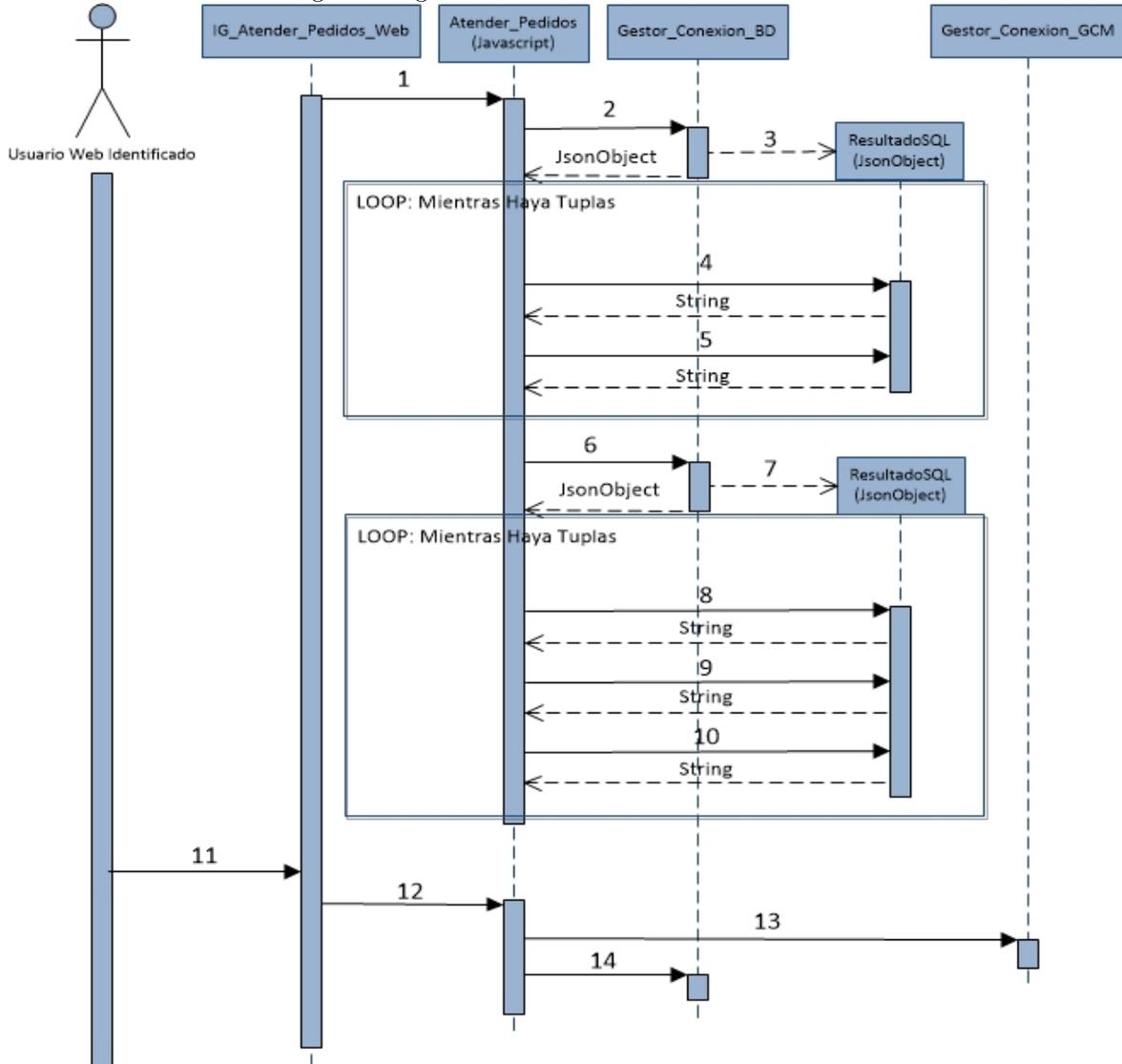
93. Figura: Diagrama de Secuencia Web: Atender Pedido



1: cargarPedidosPendientes(idRestaurante);
 2: obtenerPedidosPendientesDeAtencion(idRestaurante) : JsonObject;
 3: new();
 [LOOP Mientras haya tuplas]
 4: getString("nombreArticulo") : String;
 5: getString("descripcionArticulo") : String;
 6: obtenerPedidosPendientesDeCobro(idRestaurante) : JsonObject;
 7: new();
 [LOOP Mientras haya tuplas]
 8: getString("numeroTicket") : String;
 9: getString("nombreUsuario") : String;
 10: getString("precio") : String;
 11: El usuario pulsa el botón "Atender Pedido"
 12: atenderPedido(numer_ticket);
 13: new("Inserte el mensaje que el usuario recibirá en su dispositivo");
 14: El usuario Inserta el mensaje y pulsa en el botón "Enviar"
 15: enviarMensajeAlDispositivo(mensaje, nombreUsuario);
 16: marcarPedidoComoAtendido(numeroPedido);

Diagrama de Secuencia Web: Gestionar Cobro

94. Figura: Diagrama de Secuencia Web: Gestionar Cobro



1: cargarPedidosPendientes(idRestaurante);
 2: obtenerPedidosPendientesDeAtencion(idRestaurante) : JsonObject;
 3: new();
 [LOOP Mientras haya tuplas]
 4: getString("nombreArticulo") : String;
 5: getString("descripcionArticulo") : String;
 6: obtenerPedidosPendientesDeCobro(idRestaurante) : JsonObject;
 7: new();
 [LOOP Mientras haya tuplas]
 8: getString("numeroTicket") : String;
 9: getString("nombreUsuario") : String;
 10: getString("precio") : String;
 11: El usuario pulsa el botón "Cobrar Pedido"
 12: cobrarPedido(numer_ticket);
 13: enviarMensajeAlDispositivo("Su pedido ha sido atendido", nombreUsuario);
 14: marcarPedidoCobrado(numeroPedido);

ANEXO III.- Manual de Usuario Android

Instalación

La instalación de la aplicación deberá de realizarse manualmente a través del dispositivo, dado que no se encuentra disponible en Google Play.

Para proceder con la instalación tendremos que copiar el fichero "FastService.apk" a nuestro dispositivo, para ello podremos transferirla a través del cable USB o bien a través de algún programa de almacenamiento en la nube.

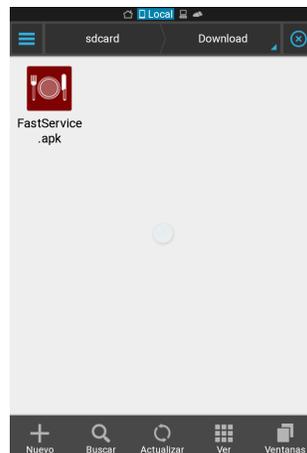
Una vez la tengamos en nuestro dispositivo, deberemos asegurarnos que el dispositivo permite la instalación de aplicaciones que no provengan de Google Play, para ello nos dirigimos a los Ajustes > Seguridad > Marcar la opción de "Orígenes desconocidos".

95. Figura: Manual Usuario Android: Instalación 1



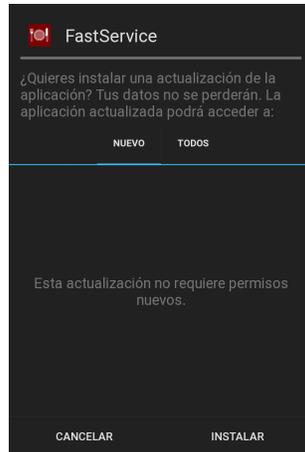
A continuación deberemos utilizar nuestro gestor de archivos preferido (en mi caso ES File Explorer) y dirigirnos a la carpeta en la cual hemos introducido la apk.

96. Figura: Manual Usuario Android: Instalación 2



Pulsamos en la aplicación y procedemos con su instalación. Nos aparecerá los permisos los cuales hace uso la aplicación, solo tendremos que pulsar en el botón "Instalar".

97. Figura: Manual Usuario Android: Instalación 3



Una vez la instalación haya finalizado ya tendremos disponible la aplicación FastService en nuestro dispositivo.

Manual

98. Figura: Manual Usuario Android: Pantalla Principal



En la pantalla principal (figura 98) encontramos dos opciones:

1. Registrarse

99. Figura: Manual Usuario Android: Registro



Nuevo usuario

Nombre de usuario:

Nombre:

Apellido:

Contraseña:

Confirmar Contraseña:

Email:

Crear Usuario

En esta pantalla debemos rellenar los datos que se piden a continuación y pulsar el botón de crear usuario, una vez hecho esto nos devolverá nuevamente a la pantalla principal (Figura 98) para proceder a iniciar sesión.

2. Iniciar sesión, si los datos de usuario son correctos al pulsar iniciar sesión, nos aparecerá una pantalla con breves instrucciones de como comenzar a usar la aplicación (Figura 100).

100. Figura: Manual Usuario Android: Instrucciones



Estando en esta pantalla de instrucciones (Figura 100) podemos realizar dos acciones:

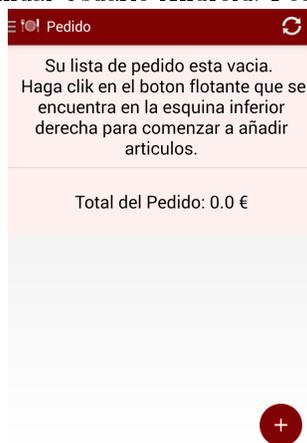
1. *Escanear Código QR*

101. Figura: Manual Usuario Android: Escanear QR



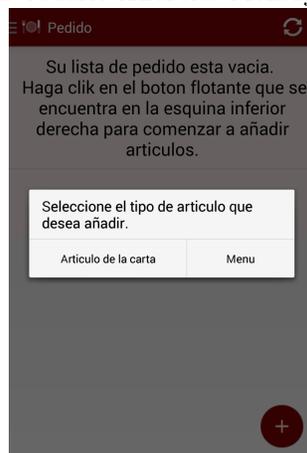
Al escanear el código QR nos mostrará la pantalla en la cual podremos comenzar a realizar nuestro pedido del restaurante (Figura 102).

102. Figura: Manual Usuario Android: Pedido sin Artículos



Es esta pantalla tenemos la opción de pulsar el botón con el signo '+' y de esta manera nos aparecerá un mensaje (Figura 103) en el cual nos permite seleccionar si queremos un artículo de la carta o un menú.

103. Figura: Manual Usuario Android: Mensaje al Realizar Pedido



- a) Si seleccionamos artículos de la carta, nos aparecerá una pantalla (Figura 104) donde encontraremos una lista de desplegables, en la cual se puede seleccionar el artículo que se quiera.

104. Figura: Manual Usuario Android: Lista Carta



Si desplegamos cualquier elemento de la lista nos aparecerán lo que podemos seleccionar (Figura 105).

105. Figura: Manual Usuario Android: Lista Carta Desplegada



Si seleccionamos uno de ellos, por ejemplo: Macarrones con Tomate, nos llevará a una pantalla (Figura 106) en la que podemos observar una imagen y descripción de dicho artículo y un botón para añadirlo a nuestra lista.

106. Figura: Manual Usuario Android: Detalles Artículo



En la pantalla de detalle podríamos volver a atrás o pulsar el botón de añadir. Si es pulsado el botón de añadir, nos aparecerá un mensaje en el que nos indicará que el artículo fue añadido a nuestro pedido (Figura 107) y luego nos devolverá a la lista de desplegados (Figura 105) para seguir seleccionando artículos si se quiere.

107. Figura: Manual Usuario Android: Mensaje de Elemento Añadido al Pedido



Si no se quieren seleccionar mas artículos, simplemente volvemos atrás y nos devuelve a el pedido donde podemos observar lo que hemos pedido (Figura 108).

108. Figura: Manual Usuario Android: Pedido con Artículos



- b) Si seleccionamos la opción de menú, nos irán apareciendo cada uno de los platos con un menú desplegable para seleccionar entre la cantidad que tendremos en cada uno de ellos (Figura 109, Figura 110, Figura 111, Figura 112) al seleccionar cada uno de estos podremos observar los detalles (Figura 106) y se podrá pulsar el botón de añadir el cual hará que el sistema nos muestre el siguiente plato a seleccionar o el botón de atrás el cual nos llevará nuevamente a la lista a seleccionar.

109. Figura: Manual Usuario Android: Menú-Primer Plato



110. Figura: Manual Usuario Android: Menú-Segundo Plato



111. Figura: Manual Usuario Android: Menú-Postre

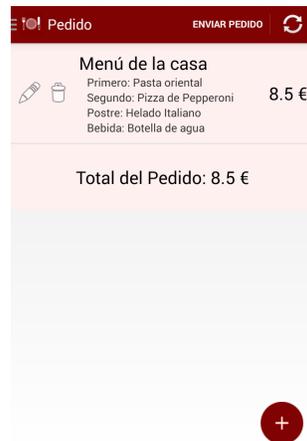


112. Figura: Manual Usuario Android: Menú-Bebida



Una vez acabado de seleccionar todos los elementos que entran en el menú el sistema nos llevará de nuevo a la pantalla de pedido (Figura 113) donde observaremos lo que hemos seleccionado.

113. Figura: Manual Usuario Android: Pedido de Menú



Una vez que tenemos seleccionados todos los elementos del pedido (Figura 114) podemos modificar elementos del menú, eliminar algún artículo de la carta, seguir seleccionando elemento o enviar el pedido.

114. Figura: Manual Usuario Android: Pedido



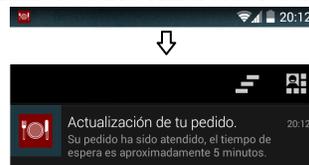
Una vez enviado el pedido los elementos quedarán pendientes (Figura 115) una vez atendido llegara una notificación en la que informará en cuanto tiempo se tendrá dicho pedido en la mesa (Figura 116).

Una vez pulsado el botón de enviar pedido, los elementos del pedido ya no podrán ser modificados ni eliminados, sin embargo se podrán seguir seleccionando elementos y enviando el pedido de los nuevos elementos (Figura 117).

115. Figura: Manual Usuario Android: Pedido Atendido



116. Figura: Manual Usuario Android: Notificación de Pedido



117. Figura: Manual Usuario Android: Artículos nuevos y Pendientes



Después de haber realizado nuestros pedidos satisfactoriamente y culminar nuestra estancia podemos solicitar pagar pedido, este botón nos aparecerá una vez que ya todos los elementos del pedido se encuentren en estado de atendido (Figura 118, Figura 119).

118. Figura: Manual Usuario Android: Pedido Atendido Boton Pagar Activo

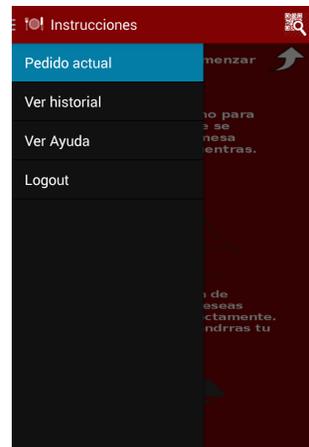


119. Figura: Manual Usuario Android: Notificación Después de Pulsar Pagar



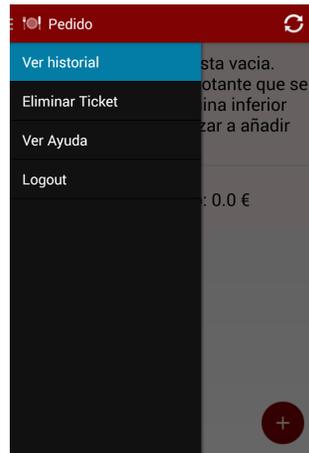
2. Desplegar el menú

120. Figura: Manual Usuario Android: Menú Desplegable



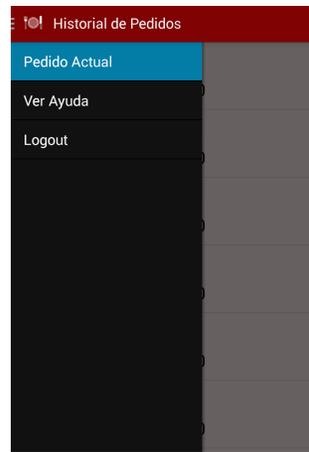
Este menú puede ser usado durante la mayoría de las pantallas que tenemos en la aplicación, a través de el se puede navegar entre los pedidos actuales, el historial de pedidos (Figura 120) , la ayuda (Figura 125) que proporciona la aplicación y cerrar sesión.

121. Figura: Manual Usuario Android: Menú Desplegable en Pedido



- a) Si nos encontramos en la pantalla de pedido y desplegamos el menú (Figura 122) se podrá:
- Ver el historial (Figura 123).
 - Eliminar Ticket, siempre que no halla ningún elemento seleccionado en la lista. Esta opción nos llevaría nuevamente a la pantalla de instrucciones (Figura 100).
 - Ver la ayuda (Figura 125).
 - Cerrar sesión.

122. Figura: Manual Usuario Android: Menú Desplegable en Historial



- a) Si nos encontramos en la pantalla de historial y desplegamos el menú (Figura 122) se podrá;
- volver al pedido actual, ver la ayuda (Figura 125) o cerrar sesión.

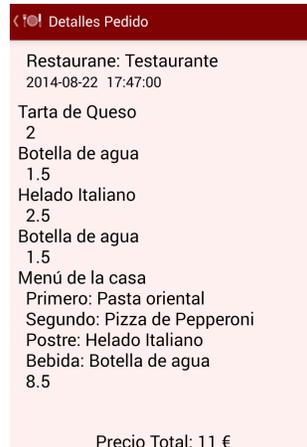
123. Figura: Manual Usuario Android: Historial



Restaurante	Fecha y Hora
Testaurante	2014-08-13 12:18:06
Testaurante	2014-08-13 14:50:45
Testaurante	2014-08-13 15:03:24
Testaurante	2014-08-13 15:26:34
Testaurante	2014-08-13 15:38:11
Testaurante	2014-08-13 16:34:21

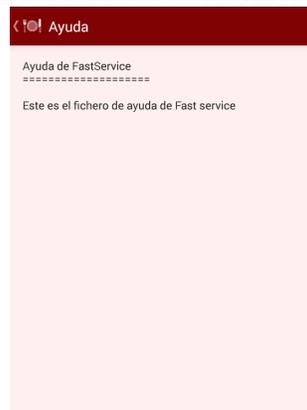
Al pulsar en un elemento del historial podrás obtener acceso a los detalles del pedido (Figura 124).

124. Figura: Manual Usuario Android: Detalle Pedido en Historial



Item	Cantidad	Precio
Restaurante: Testaurante		
2014-08-22 17:47:00		
Tarta de Queso	2	
Botella de agua	1.5	
Helado Italiano	2.5	
Botella de agua	1.5	
Menú de la casa		
Primero: Pasta oriental		
Segundo: Pizza de Pepperoni		
Postre: Helado Italiano		
Bebida: Botella de agua		
8.5		
Precio Total: 11 €		

125. Figura: Manual Usuario Android: Ayuda



Contenido
Ayuda de FastService *****
Este es el fichero de ayuda de Fast service

ANEXO IV.- Manual de la web de administración

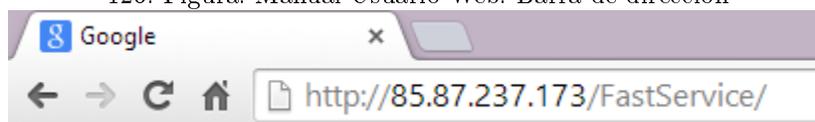
Instalación

La aplicación web no requiere ningún tipo de instalación, dado que se encuentra alojada en un servidor privado.

Para poder acceder a esta es necesario hacerlo a través de un navegador web. Para ello solo hay que introducir la siguiente dirección en la barra de direcciones del navegador:

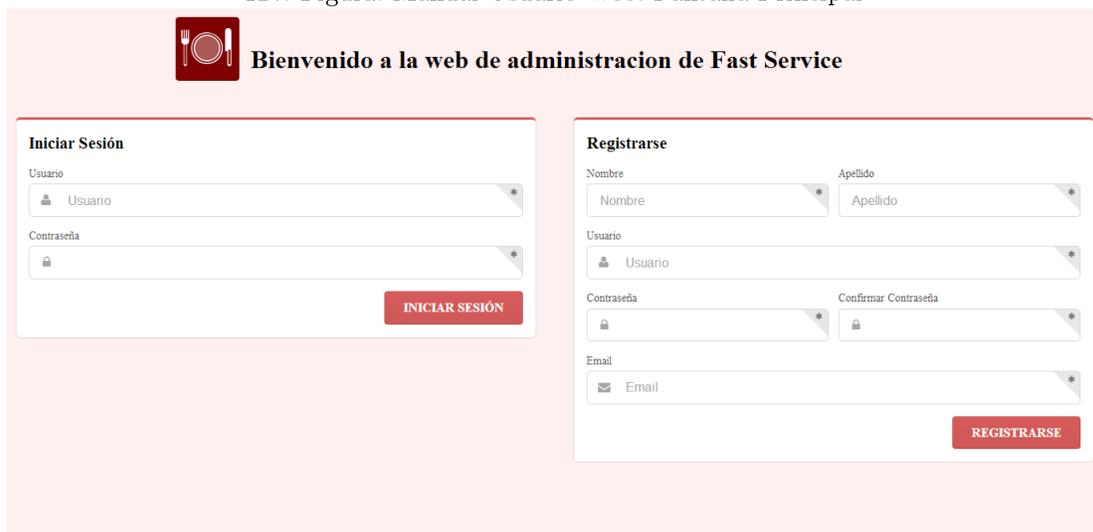
- <http://85.87.237.173/FastService/>

126. Figura: Manual Usuario Web: Barra de dirección



Manual

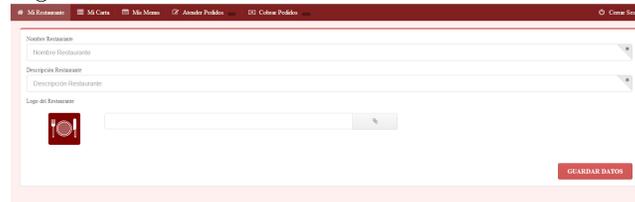
127. Figura: Manual Usuario Web: Pantalla Principal

The screenshot shows the main page of the Fast Service web administration interface. At the top left is a red square icon with a white fork and knife. To its right is the text "Bienvenido a la web de administracion de Fast Service". Below this, there are two main sections: "Iniciar Sesión" and "Registrarse". The "Iniciar Sesión" section has two input fields: "Usuario" and "Contraseña", both with a lock icon on the left and an asterisk on the right. Below these fields is a red button labeled "INICIAR SESIÓN". The "Registrarse" section has four input fields: "Nombre" and "Apellido" (with an asterisk on the right), "Usuario" (with a lock icon on the left and an asterisk on the right), "Contraseña" (with a lock icon on the left and an asterisk on the right), and "Confirmar Contraseña" (with a lock icon on the left and an asterisk on the right). Below these fields is a red button labeled "REGISTRARSE". At the bottom of the "Registrarse" section is an "Email" input field with an envelope icon on the left and an asterisk on the right.

En la *Pantalla Principal* (Figura 127) de la aplicación web para administradores podemos encontrar dos acciones, registrarse en caso de que sea su primera visita a la aplicación o iniciar sesión.

En el caso del registro, una vez rellenados los datos y pulsado el botón de registrarse, nos mostrará un mensaje de que se fue registrado correctamente y refrescara la página de esta manera ya podríamos acceder a iniciar sesión.

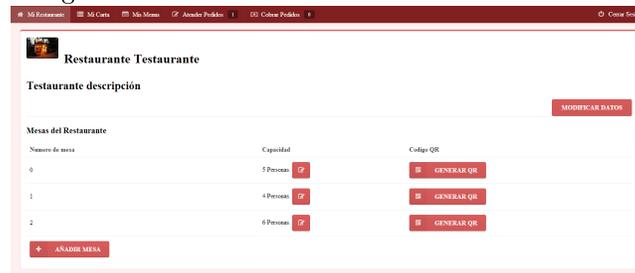
128. Figura: Manual Usuario Web: Editar Datos del Restaurante



Una vez iniciado sesión nos encontraremos con la pantalla para rellenar los datos de nuestro restaurante (Figura 128), una vez rellenados los datos, podremos comenzar a navegar entre las distintas opciones que nos ofrece el menú superior para añadir, modificar y eliminar información referente al restaurante y platos que se sirvieran en el restaurante, así como también se podrá gestionar pedidos y pagos.

1. Pestaña Mi Restaurante

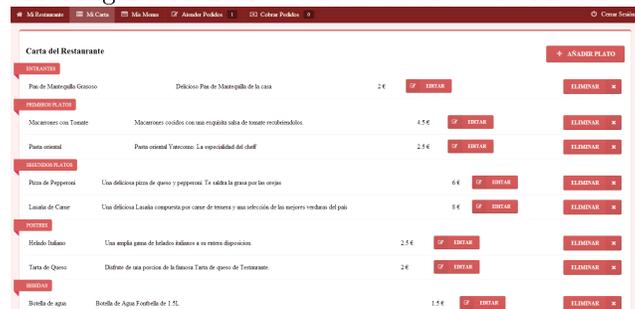
129. Figura: Manual Usuario Web: Pestaña Mi Restaurante



En la pestaña *Mi Restaurante* (Figura 129), tendremos la opción de modificar los datos del restaurante (Figura 128), así como también podremos añadir mesas indicando la cantidad de personas de las que dispondrá dicha mesa, se podrá modificar la cantidad de personas de una mesa y generar su correspondiente código QR, pero no se podrá eliminar ninguna mesa.

2. Pestaña Mi Carta

130. Figura: Manual Usuario Web: Pestaña Mi Carta



En la pestaña *Mi Carta* (Figura 130) tendremos la opción de añadir platos a la carta del restaurante (Figura 131), así como de editar (Figura 131) y eliminar los que ya se han añadido.

131. Figura: Manual Usuario Web: Añadir/Editar Plato

3. Pestaña Mis Menús

132. Figura: Manual Usuario Web: Pestaña Mis Menús

Menú de la casa	El típico menú de la casa	8,5€	ELIMINAR	EDITAR
Menú Indo-Frances	Menú que incluye platos tanto de la cocina Francesa como de la comida Asiática	18€	ELIMINAR	EDITAR
precio nombre	precio descripción	3€	ELIMINAR	EDITAR

En la pestaña *Mis Menús* (Figura 132) tendremos la opción de añadir un nuevo menú (Figura 133) seleccionando los artículos que ya tenemos en carta, una vez añadido un menú se podrá modificar o eliminar.

133. Figura: Manual Usuario Web: Añadir Menú

4. Pestaña Atender Pedidos

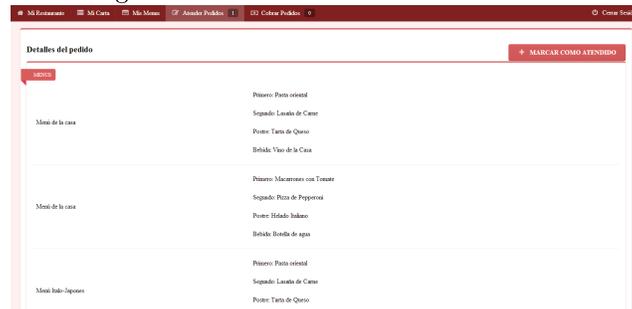
134. Figura: Manual Usuario Web: Pestaña Atender Pedidos

Pedido	Menú	Estado
1	0	listo

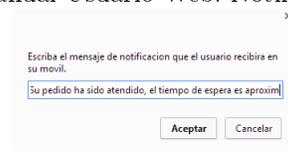
En la pestaña *Atender Pedidos* (Figura 134) lo primero que se puede observar es que la propia pestaña te informa de el número de pedidos que se encuentran pendientes ya que esta pantalla

nos muestra los pedidos que han realizado los usuarios a través de la aplicación móvil y aun no han sido atendidos, en ella encontraremos el número de pedido, la mesa y el nombre del usuario que ha realizado el pedido, además de poder observar los detalles de dicho pedido (Figura 135) para poder informar a cocina del mismo y en poder marcar como atendido, esto generará una pantalla en la que se le podrá enviar un mensaje al usuario (Figura 136) , una vez marcado como atendido este pedido desaparecerá de la lista y disminuirá el número que aparece en la pestaña.

135. Figura: Manual Usuario Web: Detalles Pedido



136. Figura: Manual Usuario Web: Notificación a Usuario



5. Pestaña Cobrar Pedidos

137. Figura: Manual Usuario Web: Pestaña Cobrar Pedidos



En la pestaña *Cobrar Pedidos* (Figura 137) también observamos que tiene el número de los elementos que se encuentran pendientes por cobrar, una vez en esta pantalla nos muestra el número de pedido, el nombre del usuario que lo ha realizado, el precio al que asciende y un botón en el que podemos marcarlo como pagado. Una vez marcado como pagado el número de la pestaña disminuirá y este elemento disminuirá de nuestra lista.