**CAPITULO 6-DESARROLLO DEL PROTOTIPO DE REALIDAD AUMENTADA.**

*Análisis del sistema*

Fuentes de Requerimientos:

1. Observación: Se observo en detalle la forma de atención que brinda el restaurant a sus clientes.
2. Entrevista formal: Se realizaron unas series de preguntas para aclarar y obtener los requerimientos funcionales y no funcionales que debe cumplir el prototipo.
3. Prototipo funcional: Se implemento un prototipo funcional para aclarar los requerimientos funcionales y no funcionales.
4. Análisis de documentación existente: Se obtuvo la documentación que se utiliza actualmente en el proceso de negocio del restaurant.
5. Tormentas de ideas: Se hicieron reuniones informales donde se aplico la técnica de tormentas de ideas para definir al sistema en cuestión.

Técnica de Entrevista

Primera entrevista realizada al Cliente

* Entrevistado: Pablo Chávez
* Entrevistadores: Carrizo Fernando Martin y Vale Walter Jesús.
* Fecha: 30 de Septiembre de 2014.
* Lugar: Confitería Charlotte.
* Objetivo: Conocer cómo funciona el negocio, como brinda sus servicios, que tipo de información se maneja, como se transmite la información entre los empleados y que expectativas tiene de un sistema informático.

*Preguntas de la Entrevista*:

1. ¿ Como es el proceso de atención de un cliente?.
2. ¿Cómo brindan el servicio a personas con discapacidad auditiva?.¿Su negocio está preparado para brindar un servicios de estas características?.
3. ¿El tipo de información se genera al finalizar la atención de un cliente? Por ejemplo: archivos digitales, documentación u otro tipo de contabilidad del restaurante.
4. ¿Cómo considera usted la forma en que actualmente se maneja la Información?.
5. ¿Le gustaría que el manejo de la información fuera automatizada, explique?.
6. ¿Haciendo referencia al chef del restaurante, como obtiene la información para la elaboración de los platos? ¿Cómo se la transmite esa información? Por ejemplo: de persona a persona, en papel, etc.
7. ¿Qué datos le gustaría guardar de su negocio de forma automática?.
8. ¿Qué datos considera necesarios a incluir en los reportes?.
9. ¿Si usted tuviera un sistema para automatizar el proceso de atención de un cliente, que personas tendrían acceso al sistema? Por ejemplo: empleados, familiares, etc.
10. ¿Cuenta con algún sistema de apoyo en su negocio?.
11. ¿Se siente a gusto con la carta gourmet del restaurante? ¿Planea hacer cambios futuros sobre esta?.
12. ¿Cuenta con dispositivos móviles y/o PC en su negocio?.
13. ¿Cuenta con conexión a internet wifi?.

Respuestas del usuarios:

1. Cuando el cliente llega el mozo le presenta la carta y espera su pedido. Según si hay mucha gente, el lo anota o no. Después me pasa el pedido, yo lo anoto y después se manda a la cocina.
2. EL negocio no está preparado para atender gente hipoacúsicas. Pero si para ayudar a gente con discapacidades motoras, como gente en silla de rueda, muletas ,etc.
3. No se trabaja con datos digitales en el negocio. Pero se lleva un registro de la contabilidad día a día en papel.
4. La forma en la que se lleva la contabilidad debería ser más ágil, creo q con un software se lograría agilidad en la información.
5. Me gustaría tener un software que lleve la contabilidad del negocio.
6. Los pedidos se hacen llegar al chef en forma verbal, y escrita si hay mucha gente.
7. Los datos q se deberían automatizar son: de las compras, las bebidas, los materiales para los platos, información de proveedores.
8. Los reportes deberían tener información sobre las compras y ventas, y deberían ser generados semanalmente, por temporada y por años. Mostrando la información en graficas de barras , pasteles, etc.
9. Los que tendrían acceso al sistema serian: el propietario, su pareja, y un empleado con restricciones de caja de ventas.
10. No se cuenta con ningún tipo de sistema de apoyo.
11. El formato de la carta se mantiene pero está planeado ampliarla.
12. Si, pero le gustaría comprar una PC o Tablet.
13. Si, cuenta con conexión a internet wifi.

Segunda entrevista realizada al Cliente

Mediante estas fuentes de requerimientos se pudo elaborar un enunciado acerca del sistema a desarrollar

Se pretende desarrollar un sistema de pedidos móvil para un restaurant que será utilizado por personas sordas e hipoacúsicas, sin ayuda de un intermediario como un mozo. El restaurant es un negocio que vende platos elaborados, bebidas, postres y también un sector de cafetería. El negocio utiliza una carta gourmet tradicional y cuenta con un mozo para toma el pedido a los clientes.

El sistema tendrá como base una carta gourmet con realidad aumentada que será construida por el dispositivo móvil. Este dispositivo debe permitir a los clientes poder navegar por la carta gourmet y además visualizar los diferentes platos 3D. El sistema debe funcionar en un celular con sistema operativo Android 2.3 o superior.

El cliente podrá realizar un alta, modificación y consulta de un pedido a través de su teléfono celular. El sistema debe estar vinculado a una conexión de red wifi para realizar todas las operaciones antes mencionadas.

El administrador debe realizar un login para poder ingresar al sistema. Una vez iniciado, podrá consultar los pedidos realizados por los clientes e imprimir esta consulta mediante un reporte. Este impreso será utilizado por el chef para la elaboración del plato. El administrador podrá ver los pedidos realizados entre un intervalo de fechas definidos. Además podrá modificar el precio del plato en la carta gourmet en caso que se requiera. Este cambio será reflejado en la carta gourmet de forma dinámica utilizando tecnología de realidad aumentando.

Los clientes a través del dispositivo móvil podrán consultar los diarios digitales más visitados. El administrador del sistema podrá dar de alta un nuevo diario y eliminarlo en caso que no se lo requiera. El acceso a los diarios digitales deberá ser implementado con realidad aumentada.

El cliente podrá ver un video de la recomendación del chef u otros videos de entretenimiento. El administrador del sistema podrá agregar, cargar y eliminar el video en caso de ser necesario. La cámara del dispositivo deberá generar un enlace mediante realidad aumentada, que al momento de presionarlo, este deberá redireccionar automáticamente a la página de youtube.

Del enunciado anterior, se obtuvo los requerimientos funcionales y no funcionales del sistema a desarrollar:

Requerimientos funcionales: Estos requerimientos son las declaraciones de los servicios que proveerá el sistema.

* El dispositivo debe permitir a los clientes poder navegar por la carta gourmet y además visualizar los diferentes platos 3D.
* Además el administrador podrá modificar el precio del plato en la carta gourmet.
* El cliente podrá realizar una Alta de pedido.
* El administrador podrá imprimir los pedidos registrados que serán utilizados por el chef para su posterior elaboración.
* El cliente podrá realizar una Modificación del pedido.
* El cliente podrá realizar una consulta del pedido.
* El sistema debe permitir una Login del administrador.
* El administrador podrá generar reportes a partir de las consultas registradas entre un intervalo de fechas.
* Cliente podrá seleccionar un video de una lista y reproducirlo.
* El administrador podrá agregar un video.
* El administrador del sistema podrá eliminar un video de la recomendación del chef.
* El cliente podrá visitar los diarios digitales más populares.
* El administrador podrá agregar un diario.
* El administrador cargar un video
* El administrador podrá eliminar un diario.
* La cámara del dispositivo debe generar un enlace mediante realidad aumentada.

Requerimientos no funcionales: Estos requisitos especifican varios parámetros operacionales que definen el entorno en el cual existe el sistema. Estos son criterios que definen niveles de desempeño, escalabilidad, requisitos de seguridad, copias de seguridad y otros requisitos operacionales.

* Sistema va a utilizar una conexión de red wifi para realizar todas las operaciones relacionadas con los pedidos.
* La carta gourmet va a ser implementado con realidad aumentada.
* El lenguaje de programación según el análisis de tecnologías va a ser C#.
* El sistema debe contar con un ordenador con las siguientes características:

|  |  |  |
| --- | --- | --- |
| Requerimientos | Mínimos | Recomendados |
| Memoria RAM | 512 MB | 1 GB |
| Espacio en disco duro | 40 MB de espacio disponible | 200 MB de espacio disponible |
| Procesador | Intel Pentium o superior | |
| Sistema operativo | Windows XP con SP2 o superior | |
| Periféricos | Teclado, mouse, tarjeta de red inalámbrica. | |

* El sistema debe contar con un dispositivo móvil con las siguientes características:

|  |  |  |
| --- | --- | --- |
| Requerimientos | Mínimos | Recomendados |
| Resolución de pantalla | 240 x 320 pixeles | 480 x 800 pixeles |
| Memoria interna | 1 GB | 2 GB |
| Cámara | 3 MP | 5MP |
| Memoria RAM | 512 MB o superior | |
| Sistema operativo | Android 2.3 o superior | |

* El ingreso al sistema va a contar con un solo tipo de perfil de usuario.
* El sistema debe permitir el reconocimiento del marcador en menos de 5 segundos.
* El sistema debe generar los modelos 3D en menos de 3 segundos.

**Especificación de requerimiento**

Según el análisis de los requerimientos, el prototipo está formado por dos sistemas relacionados por una base de datos:

* Sistema de Gestión de Pedido: Este sistema será usado por el administrador para gestionar los pedidos realizados por los clientes. El mismo se ejecutara como una aplicación de escritorio en un ordenador ubicado en el local comercial.

CU usuaruioAdministrador.emf

Figura 6.1 Diagrama de Caso de Uso del Sistema de Gestión de pedido

* Sistema de Atención Aumentada: Este sistema se ejecutara en un dispositivo móvil para tomar los pedidos del cliente utilizando tecnología de realidad aumentada. El cliente podrá visualizar los modelos 3D de los platos a consumir utilizando como marcador la carta gourmet.

CU usuarioCliente.emf

Figura 6.2 Diagrama de Caso de Uso del Sistema de Atención Aumentada

Detalle de Caso de Uso del Sistema de Gestión de Pedido

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Login. | |
| Descripción: | El administrador debe logearse para tener acceso al sistema. | |
| Actor: | Usuario Administrador. | |
| Precondición: | El Administrador debe existir en el sistema. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El administrador ingresa su usuario y contraseña | |  |
| 1. El administrador presiona el botón “Ingresar”. | |  |
| 1. El sistema verifica el usuario en la base de datos. | | * 1. El sistema devuelve un mensaje con la leyenda “El usuario o contraseña es incorrecto”   2. Regresa al Paso 1 |
| 1. El administrador ingresa a la ventana principal. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Consultar Pedido | |
| Descripción: | El administrador realiza una consulta de los pedidos realizados por los clientes | |
| Actor: | Usuario Administrador | |
| Precondición: | El Administrador debe iniciar sesión en el sistema. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El administrador elige la pestaña “Pedidos” en la ventana principal. | | 1.1 El administrador ingresa una fecha de inicio y final.  1.2 El sistema realiza una búsqueda según el intervalo de fecha definido |
| 1. El sistema carga en una tabla los pedidos realizados. | |  |
| 1. El administrador selecciona un pedido y presiona el botón “Detalle”. | |  |
| 1. El sistema carga en una tabla el detalle del pedido seleccionado previamente. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Generar Pedido | |
| Descripción: | El administrador genera un reporte en formato pdf. | |
| Actor: | Usuario Administrador | |
| Precondición: | El Administrador debe realizar el caso de uso “Consultar Pedido” | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El administrador presiona el botón “Generar Pedido” | |  |
| 1. El sistema genera un reporte en formato pdf. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Agregar Video | |
| Descripción: | Le permite al administrador agregar un nuevo video de las recomendaciones del chef o de las promociones ofrecidas. | |
| Actor: | Usuario Administrador | |
| Precondición: | El Administrador debe estar logeado en el sistema. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El administrador presiona la pestaña “Videos” en la ventana principal del Sistema. | |  |
| 1. El administrador presiona el botón “Agregar” | |  |
| 1. El sistema despliega una venta para cargar enlace del video. | |  |
| 1. El administrador ingresa enlace de video y presiona el botón “Aceptar” | | 4.1 El administrador presiona el botón “Cancelar” |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Cargar Video | |
| Descripción: | El administrador puede cargar un video de las recomendaciones del chef u otro video de entretenimiento. | |
| Actor: | Usuario Administrador | |
| Precondición: | Debe haber por lo menos un video cargado en el sistema. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El administrador presiona la pestaña “Videos” en la ventana principal. | |  |
| 1. El sistema despliega una venta con una lista de los videos subidos a la base de datos. | |  |
| 1. El administrador selecciona un video | |  |
| 1. El administrador presiona el botón “Cargar Video” | |  |
| 1. Se actualiza el sistema con los cambios realizados. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Eliminar Video | |
| Descripción: | El administrador eliminar un video de las recomendaciones del chef u otro video de entretenimiento. | |
| Actor: | Usuario Administrador | |
| Precondición: | En el sistema debe haber por lo menos un video cargado. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El administrador presiona la pestaña “Videos” en la ventana principal. | |  |
| 1. El sistema despliega una venta con la lista de los videos subidos a la base de datos. | |  |
| 1. El administrador selecciona un video | |  |
| 1. El administrador presiona el botón “Eliminar” | |  |
| 1. Se actualiza el sistema con los cambios realizados. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Modificar Precio | |
| Descripción: | El administrador modificar el precio de un plato de la carta gourmet. | |
| Actor: | Usuario Administrador | |
| Precondición: | En el sistema debe haber por lo menos un plato cargado en el sistema. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El administrador presiona la pestaña “Menú” en la ventana principal del sistema. | |  |
| 1. El sistema carga una ventana con una lista de todos los platos cagados en el sistema. | |  |
| 1. El administrador selecciona un plato de la lista y presiona “Modificar Plato” | |  |
| 1. El sistema carga un formulario de modificación. | |  |
| 1. El administrador ingresa el precio y presiona “Aceptar” | | 5.1 Se presiona el botón cancelar |
| 1. El sistema guarda el nuevo precio en la base de datos. | |  |

Detalle de Caso de Uso del Sistema de Atención Aumentada

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Alta de pedido | |
| Descripción: | El cliente realiza un pedido mediante el dispositivo móvil. | |
| Actor: | Usuario Cliente | |
| Precondición: | Se debe generar la carta gourmet aumentada previamente. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El cliente presiona el botón “Agregar” para añadir un plato al pedido. | |  |
| 1. El cliente presiona el botón “Ver Pedidos” | |  |
| 1. El sistema desplegara una ventana con la lista de todos los platos añadidos. | |  |
| 1. El cliente presiona el botón “Enviar Pedido” | | 4.1 El cliente presiona el botón “Cancelar”. |
| 1. El sistema guarda en la base de datos el pedido. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Modificar Pedido | |
| Descripción: | Le permite al cliente modificar un pedido antes de ser enviado. | |
| Actor: | Usuario Cliente | |
| Precondición: | Debe haber por lo menos un plato añadido en el pedido. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El cliente presiona el botón “Ver Pedidos” | |  |
| 1. El sistema desplegara una ventana con la lista de los platos añadidos. | |  |
| 1. El cliente puede quitar un plato presionando el botón “X”. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Consultar Pedido | |
| Descripción: | Le permite al cliente consultar el pedido antes de ser enviado | |
| Actor: | Usuario Cliente | |
| Precondición: | Debe agregarse por lo menos un plato al pedido | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El cliente presiona el botón “Ver Pedidos” | |  |
| 1. Se despliega una ventana con los pedidos añadidos. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Consultar Diarios Digitales | |
| Descripción: | El cliente puede leer los diarios digitales mas leídos. | |
| Actor: | Usuario Cliente | |
| Precondición: | El cliente debe enfocar con la cámara al marcador de los diarios digitales. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El cliente enfoca con la cámara al marcador de diarios digitales. | |  |
| 1. El cliente presiona el enlace aumentada del diario digital. | |  |
| 1. El sistema desplegara una lista con todos los diarios digitales mas leídos. | |  |
| 1. El cliente presiona el botón del diario que necesita leer. | | 4.1 El cliente presiona el botón “Cancelar” |
| 1. El sistema redirigirá a la página del diario digital. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Consultar Video | |
| Descripción: | El cliente puede ver un video de las recomendaciones del chef u otro video de entretenimiento. | |
| Actor: | Usuario Cliente | |
| Precondición: | El sistema debe haber por lo menos un video cargado. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El cliente enfoca con la cámara al marcador de videos. | |  |
| 1. El cliente presiona el enlace aumentada del video. | |  |
| 1. El sistema genera una ventana con una lista de los videos a visitar. | |  |
| 1. El cliente presiona el botón del nombre del video | | 4.1 El cliente presiona el botón “Cancelar” |
| 1. El sistema redirige a la página de youtube. | |  |

|  |  |  |
| --- | --- | --- |
| Caso de Uso: | Navegar Carta Gourmet | |
| Descripción: | Permite al cliente poder desplazarse por la carta gourmet visualizando los diferentes platos. | |
| Actor: | Usuario Cliente | |
| Precondición: | El cliente debe enfocar en la carta gourmet para generar la carta de realidad aumenta. | |
| Flujo de Eventos | | |
| Flujo Normal | | Flujo Alternativo |
| 1. El cliente presiona el botón virtual “siguiente” | | 1.1 El cliente presiona el botón virtual “anterior” |
| 1. El sistema desplaza el selector de platos a una posición siguiente en la carta gourmet | | 2.1 El sistema desplaza el selector de platos a una posición anterior en la carta gourmet |
| 1. El sistema construye una representación 3D del plato | | Salta al paso 3 |

*Conformación del Equipo de Trabajo*

|  |  |
| --- | --- |
| ROL | PERSONA |
| ***Product Owner*** | Pablo (Dueño de restorán) |
| ***Scrum Master*** | Reinoso, Elizabeth |
| ***Scrum Team*** | Vale, Walter Jesús  Carrizo, Fernando Martin |
| ***Usuarios o Cliente*** | Amigos de la profe. |

**Diseño del Prototipo**

Arquitectura de la aplicación

Nuestra arquitectura no varía mucho de la arquitectura original de vuforia. La diferencia principal es la inclusión de una base de datos con la que actúa la aplicación, la cual contiene toda la información que muestra el menú y gestiona las transacciones de los pedidos realizados.



Figura 2.7 Arquitectura de la aplicación del proyecto presente.

Diccionario de datos

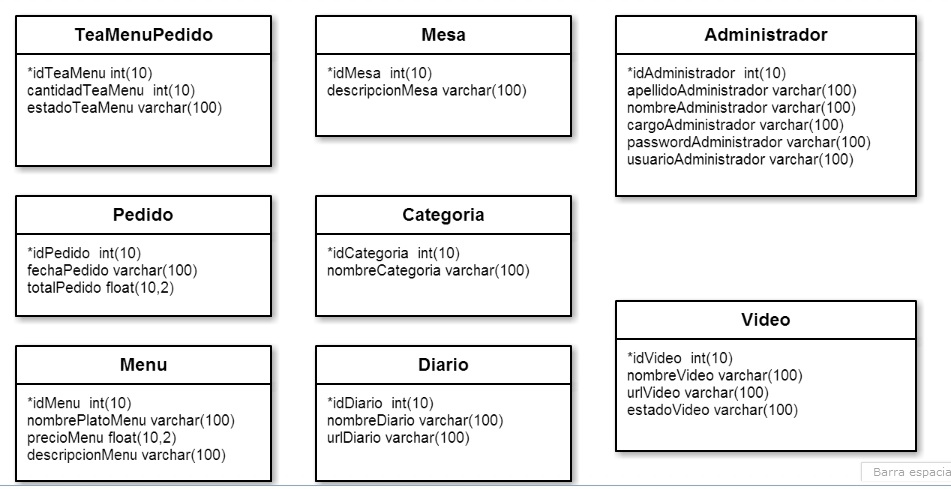


Diagrama Entidad - Relacion Global

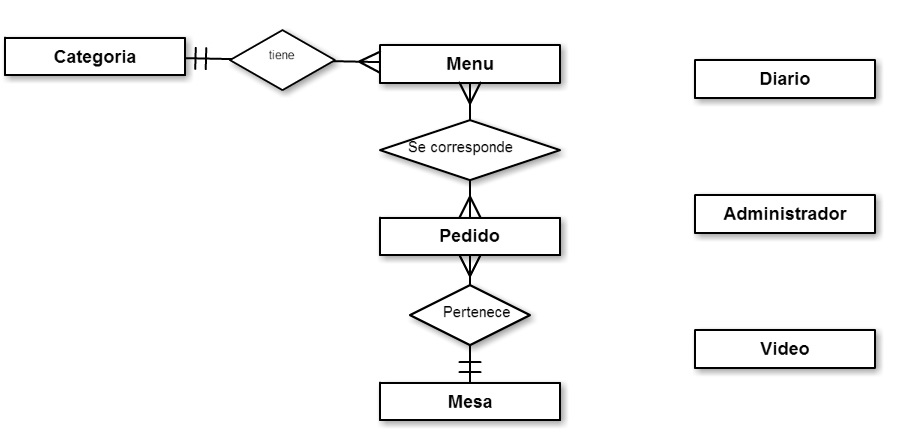


Diagrama de Martin

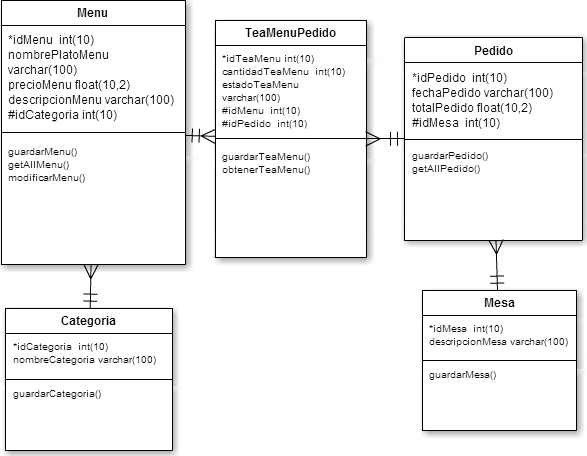
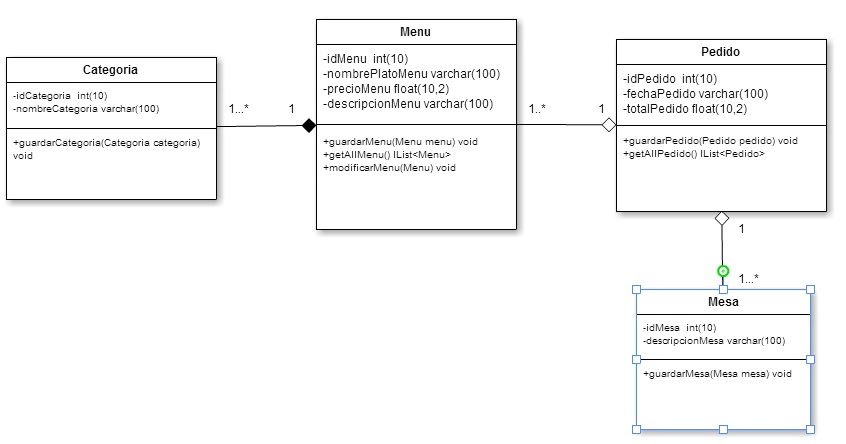
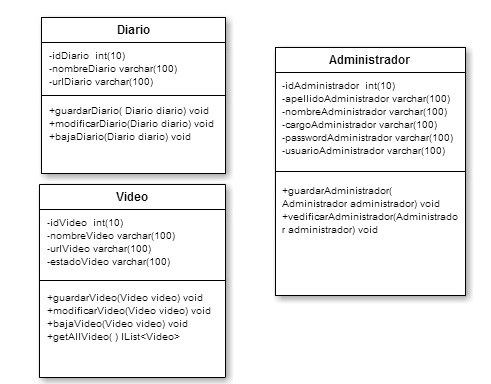


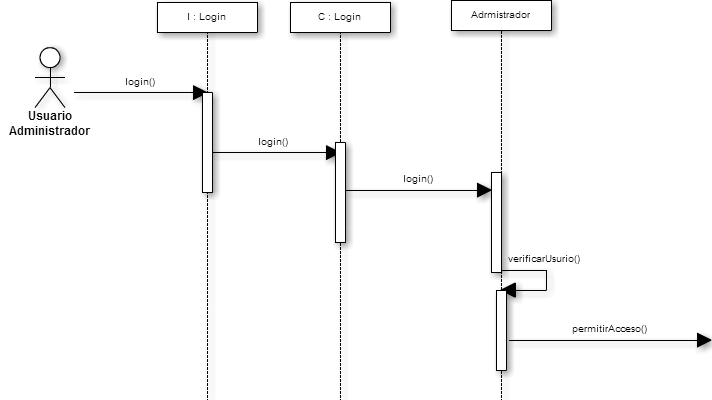
Diagrama de clases



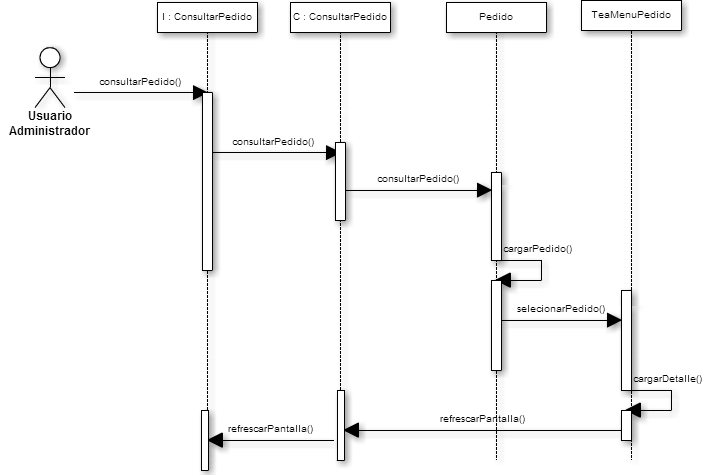


Diagramas de secuencia

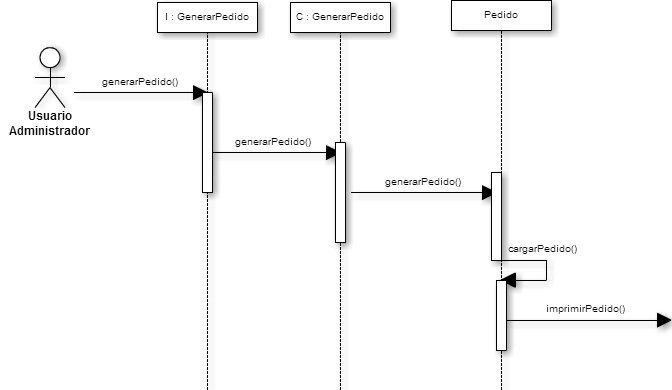
Login



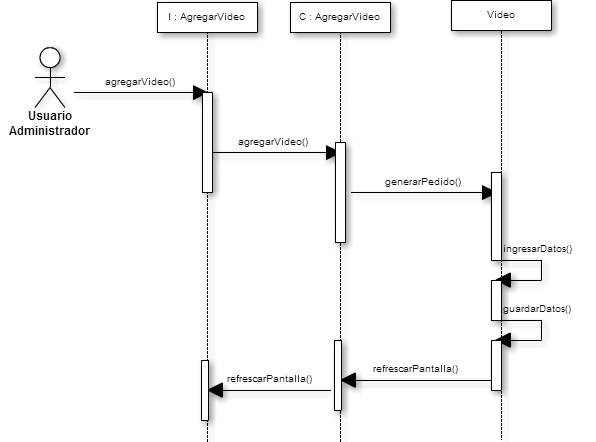
Consultar Pedido



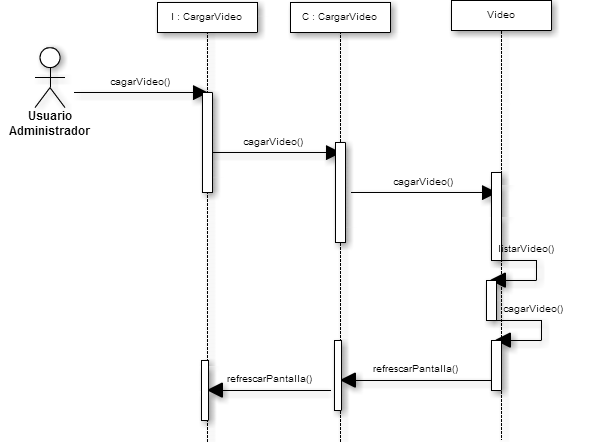
Generar Pedido



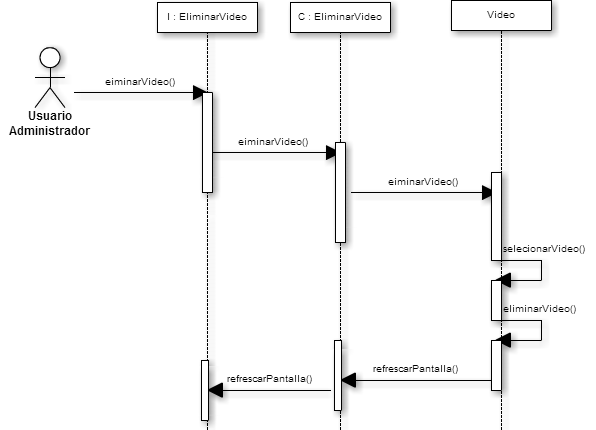
Agregar Video



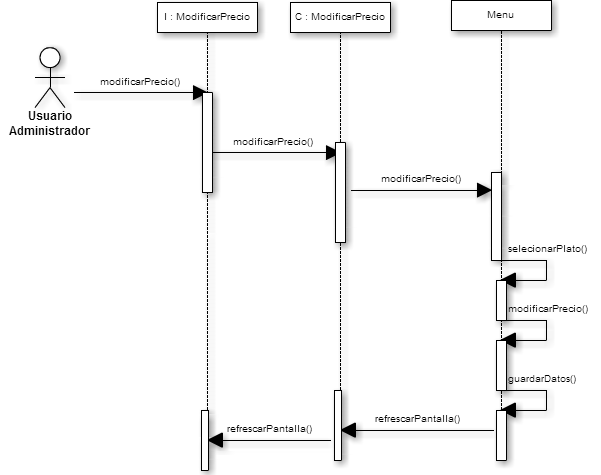
Cargar Video



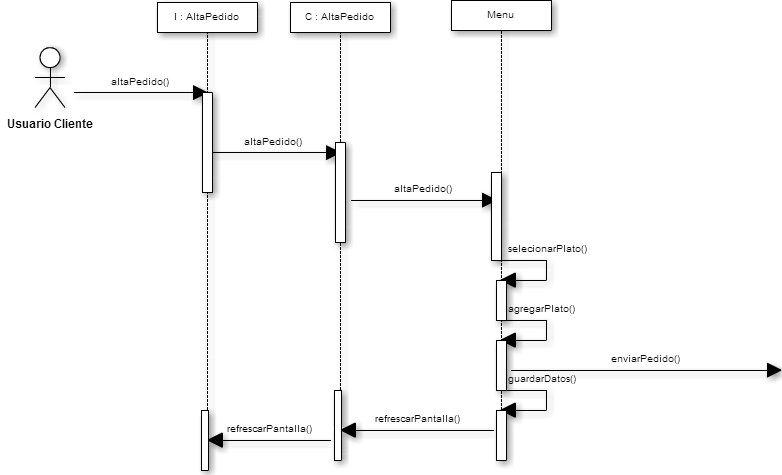
Eliminar Video



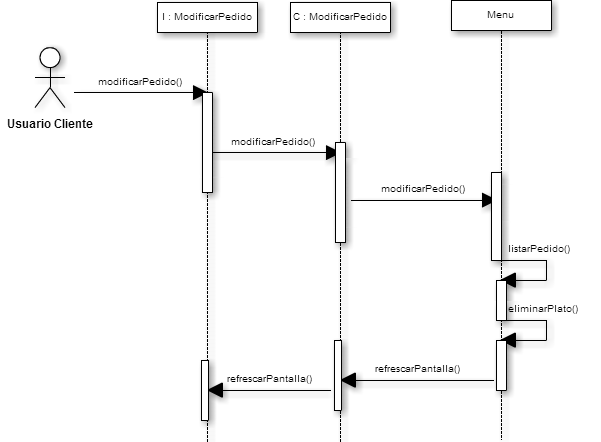
Modificar Precio



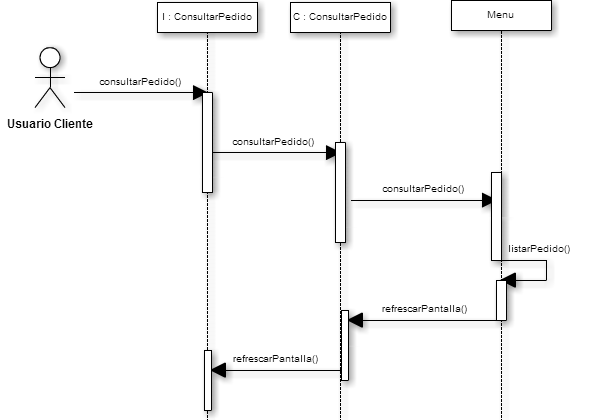
Alta de pedido



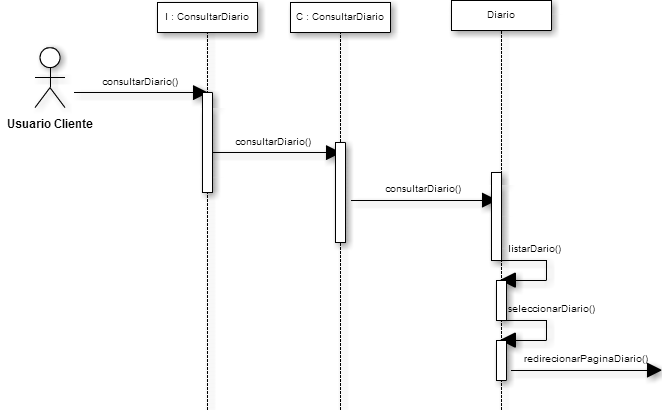
Modificar Pedido



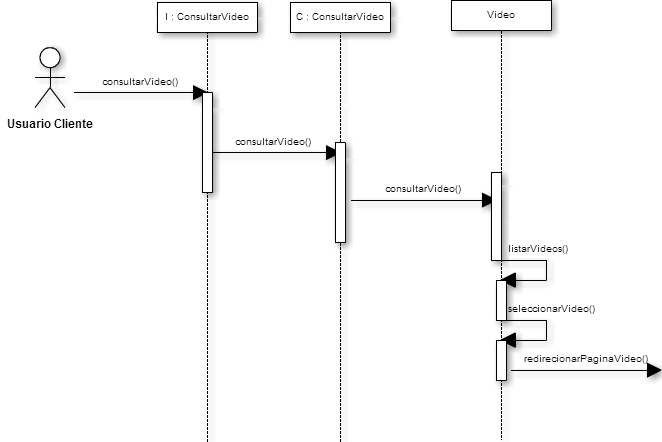
Consultar Pedido



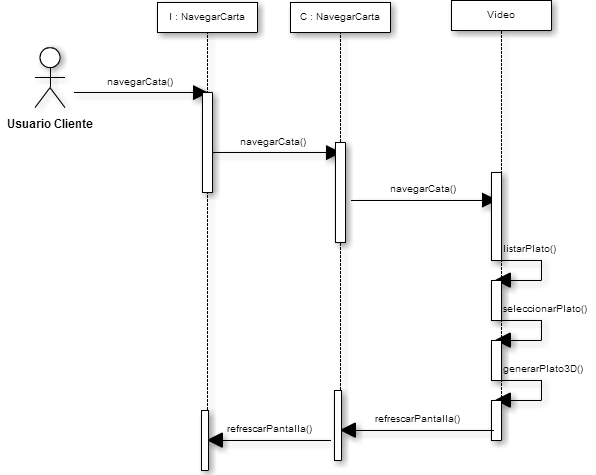
Consultar Diarios Digitales



Consultar Video



Navegar Carta Gourmet



Definición de Interfaz de usuario

Codificación

La codificación del prototipo está divido en dos sistemas, una aplicación para dispositivo móvil llamada “Sistema de Atención Aumentada” y otra para escritorio llamada “Sistema de Gestión de Pedido”.

Codificación del Sistema de Atención Aumentada

Para el desarrollo de la aplicación móvil se utiliza la herramienta Unity 3D, un motor grafico para crear aplicaciones interactivas y el framework de Realidad Aumentada Vuforia . La codificación se realiza en con lenguajes de scripts, más precisamente el lenguaje C#. A continuación se muestra el código desarrollado:

**Clase Menú**

**using** UnityEngine**;**

**using** System**.**Collections**;**

//Plantilla para la creacion de objetos de tipo menu

**public** class Menu **:** MonoBehaviour **{**

**private** int idMenu**;**

**private** string nombrePlatoMenu**;**

**private** double precionMenu**;**

**private** string descripcionMenu**;**

**private** Categoria categoria**;**

**public** Menu**(){}**

**public** Menu**(**int idMenu**,**string nombrePlatoMenu**,**double precioMenu**,**

string descripcionMenu**,** Categoria categoria**){**

**this.**idMenu**=**idMenu**;**

**this.**nombrePlatoMenu**=**nombrePlatoMenu**;**

**this.**precionMenu**=**precioMenu**;**

**this.**descripcionMenu**=**descripcionMenu**;**

**this.**categoria**=**categoria**;**

**}**

**public** int getIdMenu**(){**

**return** **this.**idMenu**;**

**}**

**public** void setIdMenu**(**int idMenu**){**

**this.**idMenu**=**idMenu**;**

**}**

**public** string getNombrePlatoMenu**(){**

**return** **this.**nombrePlatoMenu**;**

**}**

**public** void setNombrePlatoMenu**(**string nombrePlanoMenu**){**

**this.**nombrePlatoMenu**=**nombrePlanoMenu**;**

**}**

**public** double getPrecioMenu**(){**

**return** **this.**precionMenu**;**

**}**

**public** void setPrecioMenu**(**double precioMenu**){**

**this.**precionMenu**=**precioMenu**;**

**}**

**public** string getDescripcionMenu**(){**

**return** **this.**descripcionMenu**;**

**}**

**public** void setDescripcionMenu**(**string descripcionMenu**){**

**this.**descripcionMenu**=**descripcionMenu**;**

**}**

**public** Categoria getCategoria**(){**

**return** **this.**categoria**;**

**}**

**public** void setCategoria**(**Categoria categoria**){**

**this.**categoria**=**categoria**;**

**}**

**}**

**Clase Pedido**

**using** UnityEngine**;**

**using** System**.**Collections**;**

//Plantilla para la creacion de objetos de tipo Pedido

**public** class Pedido **:** MonoBehaviour **{**

**private** int idPedido**;**

**private** string fechaPedido**;**

**private** double totalPagarPedido**;**

**private** Mesa mesa**;**

**public** Pedido**(){}**

**public** Pedido**(**int idPedido**,** string fechaPedido**,** double totalPagarPedido**,**Mesa mesa**){**

**this.**idPedido**=**idPedido**;**

**this.**fechaPedido**=**fechaPedido**;**

**this.**totalPagarPedido**=**totalPagarPedido**;**

**this.**mesa**=**mesa**;**

**}**

**public** void setIdPedido**(**int idPedido**){**

**this.**idPedido**=**idPedido**;**

**}**

**public** int getIdPedido**(){**

**return** **this.**idPedido**;**

**}**

**public** void setFechaPedido**(**string fechaPedido**){**

**this.**fechaPedido**=**fechaPedido**;**

**}**

**public** string getFechaPedido**(){**

**return** **this.**fechaPedido**;**

**}**

**public** void setTotalPagarPedido**(**double totalPagarPedido**){**

**this.**totalPagarPedido**=**totalPagarPedido**;**

**}**

**public** double getTotalPagarPedido**(){**

**return** **this.**totalPagarPedido**;**

**}**

**public** void setMesa**(**Mesa mesa**){**

**this.**mesa**=**mesa**;**

**}**

**public** Mesa getMesa**(){**

**return** **this.**mesa**;**

**}**

**}**

**Clase Mesa**

**using** UnityEngine**;**

**using** System**.**Collections**;**

//Plantilla para la creacion de objetos de tipo Mesa

**public** class Mesa **:** MonoBehaviour **{**

**private** int idMesa**;**

**private** int numeroMesa**;**

**private** string descripcionMesa**;**

**public** Mesa**(){}**

**public** Mesa**(**int idMesa**,**int numeroMesa**,**string descripcionMesa**){**

**this.**idMesa**=**idMesa**;**

**this.**numeroMesa**=**numeroMesa**;**

**this.**descripcionMesa**=**descripcionMesa**;**

**}**

**public** Mesa**(**int idMesa**,**string descripcionMesa**){**

**this.**idMesa**=**idMesa**;**

**this.**descripcionMesa**=**descripcionMesa**;**

**}**

**public** int getIdMesa**(){**

**return** **this.**idMesa**;**

**}**

**public** void setIdMesa**(**int idMesa**){**

**this.**idMesa**=**idMesa**;**

**}**

**public** int getNumeroMesa**(){**

**return** **this.**numeroMesa**;**

**}**

**public** void setNumeroMesa**(**int numeroMesa**){**

**this.**numeroMesa**=**numeroMesa**;**

**}**

**public** string getDescripcionMesa**(){**

**return** **this.**descripcionMesa**;**

**}**

**public** void setDescripcionMesa**(**string descripcionMesa**){**

**this.**descripcionMesa**=**descripcionMesa**;**

**}**

**}**

**Clase Diario**

**using** UnityEngine**;**

**using** System**.**Collections**;**

//Plantilla para la creacion de objetos de tipo Diario

**public** class Diario **:** MonoBehaviour **{**

**private** int idDiario**;**

**private** string nombreDiario**;**

**private** string urlDiario**;**

**public** Diario**(){}**

**public** Diario**(**int idDiario**,** string nombreDiario**,** string urlDiario**){**

**this.**idDiario **=**idDiario**;**

**this.**nombreDiario**=** nombreDiario**;**

**this.**urlDiario**=** urlDiario**;**

**}**

**public** void setIdDiario**(**int idDiario**){**

**this.**idDiario**=** idDiario**;**

**}**

**public** int getIdDiario**(){**

**return** **this.**idDiario**;**

**}**

**public** void setNombreDiario **(**string nombreDiario**){**

**this.**nombreDiario**=** nombreDiario**;**

**}**

**public** string getNombreDiario**(){**

**return** **this.**nombreDiario**;**

**}**

**public** void setUrlDiario**(**string urlDiario**){**

**this.**urlDiario**=** urlDiario**;**

**}**

**public** string getUrlDiario**(){**

**return** **this.**urlDiario**;**

**}**

**}**

**Clase Categoría**

**using** UnityEngine**;**

**using** System**.**Collections**;**

//Plantilla para la creacion de objetos de tipo Categoria

**public** class Categoria **:** MonoBehaviour **{**

**private** int idCategoria**;**

**private** string nombreCategoria**;**

**public** Categoria**(){}**

**public** Categoria**(**int idCategoria**,**string nombreCategoria**){**

**this.**idCategoria**=**idCategoria**;**

**this.**nombreCategoria**=**nombreCategoria**;**

**}**

**public** void setIdCategoria**(**int idCategoria**){**

**this.**idCategoria**=**idCategoria**;**

**}**

**public** int getIdCategoria**(){**

**return** **this.**idCategoria**;**

**}**

**public** void setNombreCategoria**(**string nombreCategoria**){**

**this.**nombreCategoria**=**nombreCategoria**;**

**}**

**public** string getNombreCategoria**(){**

**return** **this.**nombreCategoria**;**

**}**

**}**

**Clase TeaMenuPedio**

**sing** UnityEngine**;**

**using** System**.**Collections**;**

//Plantilla para la creacion de objetos de tipo TeaMenuPedido

**public** class TeaMenuPedido **:** MonoBehaviour **{**

**private** int idTeaMenuPedido**;**

**private** Pedido pedido**;**

**private** Menu menu**;**

**private** int cantidadTeaMenuPedido**;**

**private** string estadoTeaMenuPedido**;**

**public** TeaMenuPedido**(){}**

**public** TeaMenuPedido**(**int idTeaMenuPedido**,** Pedido pedido**,** Menu menu**,**

int cantidadTeaMenuPedido**,**string estadoTeaMenuPedido**){**

**this.**idTeaMenuPedido**=**idTeaMenuPedido**;**

**this.**pedido**=**pedido**;**

**this.**menu**=**menu**;**

**this.**cantidadTeaMenuPedido**=**cantidadTeaMenuPedido**;**

**this.**estadoTeaMenuPedido**=**estadoTeaMenuPedido**;**

**}**

**public** void setIdTeaMenuPedido**(**int idTeaMenuPedido**){**

**this.**idTeaMenuPedido**=**idTeaMenuPedido**;**

**}**

**public** int getIdTeaMenuPedido**(){**

**return** **this.**idTeaMenuPedido**;**

**}**

**public** void setPedido**(**Pedido pedido**){**

**this.**pedido**=**pedido**;**

**}**

**public** Pedido getPedido**(){**

**return** **this.**pedido**;**

**}**

**public** void setMenu**(**Menu menu**){**

**this.**menu**=**menu**;**

**}**

**public** Menu getMenu**(){**

**return** **this.**menu**;**

**}**

**public** void setCantidadTeaMenuPedido**(**int cantidadTeaMenuPedido**){**

**this.**cantidadTeaMenuPedido**=**cantidadTeaMenuPedido**;**

**}**

**public** int getCantidadTeaMenuPedido**(){**

**return** **this.**cantidadTeaMenuPedido**;**

**}**

**public** void setEstadoTeaMenuPedido**(**string estadoTeaMenuPedido**){**

**this.**estadoTeaMenuPedido**=**estadoTeaMenuPedido**;**

**}**

**public** string getEstadoTeaMenuPedido**(){**

**return** **this.**estadoTeaMenuPedido**;**

**}**

**public** void aumetarCantidadTeaMenuPedido**(){**

**this.**cantidadTeaMenuPedido**=**cantidadTeaMenuPedido**+**1**;**

**}**

**public** void disminuirCantidadTeaMenuPedido**(){**

**this.**cantidadTeaMenuPedido**=**cantidadTeaMenuPedido**-**1**;**

**}**

**}**

**Administración de Servidor: Permite armar la interfaz de configuración del servidor para el ingreso de la dirección ip donde se encuentra alojado el servidor.**

**using** UnityEngine**;**

**using** System**.**Collections**;**

**using** System**.**Net**;**

**using** System**;**

/\*Permite dibujar la intefaz de configuracion de servidor\*/

**public** class AdministracionServidor **:** MonoBehaviour **{**

//GUISTyle para configuracion de interfaz.

**public** GUIStyle styleBox**;**

**public** GUIStyle styleBoxLoading**;**

**public** GUIStyle styleTextField1**;**

**public** GUIStyle styleTextField2**;**

**public** GUIStyle styleTextField3**;**

**public** GUIStyle styleTextField4**;**

**public** GUIStyle styleTitulo**;**

**public** GUIStyle styleLabel**;**

**public** GUIStyle styleBotonAceptar**;**

**public** GUIStyle styleMensajeError**;**

**public** GUIStyle styleLabelPunto**;**

//Guarda la direccion ip del servidor

**public** string ip1**;**

**public** string ip2**;**

**public** string ip3**;**

**public** string ip4**;**

//Almacena las direccion ip en memoria

**internal** int ip1Almacenada**;**

**internal** int ip2Almacenada**;**

**internal** int ip3Almacenada**;**

**internal** int ip4Almacenada**;**

//Son variables para posicionar componentes de interfaz

**private** Rect cajaConfiguracionServidor**;**

**private** Rect cajaLoading**;**

**private** Rect cajaTitulo**;**

**private** Rect cajaLabel1**;**

**private** Rect cajaLabel2**;**

**private** Rect cajaLabelPunto1**;**

**private** Rect cajaLabelPunto2**;**

**private** Rect cajaLabelPunto3**;**

**private** Rect cajaBotonAcepatar**;**

**private** Rect cajaTextField1**;**

**private** Rect cajaTextField2**;**

**private** Rect cajaTextField3**;**

**private** Rect cajaTextField4**;**

**private** Rect cajaMensajeError**;**

**private** Texture Spinner**;**

**private** string ipServidor**;**

**private** string mensajeError**;**

**private** SetearListado seterListado**;**

//Inicializacion de variables al iniciar el sistema

void Awake**()**

**{**

ipServidor**=**AppIUConstantes**.**CADENA\_VACIA**;**

mensajeError**=**AppIUConstantes**.**CADENA\_VACIA**;**

ip1 **=** AppIUConstantes**.**CADENA\_VACIA**;**

ip2 **=** AppIUConstantes**.**CADENA\_VACIA**;**

ip3 **=** AppIUConstantes**.**CADENA\_VACIA**;**

ip4 **=** AppIUConstantes**.**CADENA\_VACIA**;**

**this.**redimensionarPantalla**(**Screen**.**width**,** Screen**.**height**);**

**this.**contruirDireccionIp**();**

**}**

// Use this for initialization

void Start **()** **{**

seterListado**=**GetComponent**<**SetearListado**>();**

**this.**recuperarValoresAlmacenados**();**

**}**

// Update is called once per frame

void Update **()** **{**

**this.**redimensionarPantalla**(**Screen**.**width**,** Screen**.**height**);**

**}**

//Dibuja la interfaz de configuracion del servidor

void OnGUI**()** **{**

**if(**seterListado**.**getDesactivarLoading**()){**

GUI**.**Box**(**cajaLoading**,**AppIUConstantes**.**CADENA\_VACIA**,**

styleBoxLoading**);**

Matrix4x4 oldMatrix **=** GUI**.**matrix**;**

float thisAngle **=** Time**.**frameCount**\***4**;**

GUIUtility**.**RotateAroundPivot**(**thisAngle**,** cajaLoading**.**center**);**

GUI**.**DrawTexture**(**cajaLoading**,** Spinner**);**

GUI**.**matrix **=** oldMatrix**;**

**}**

**if(**seterListado**.**getHabilitarPantallaServidor**()){**

GUI**.**Box**(**cajaConfiguracionServidor**,**AppIUConstantes**.**CADENA\_VACIA**,**

styleBox**);**

GUI**.**Label **(**cajaTitulo**,** AppIUConstantes**.**TITULO\_SERVIDOR**,**styleTitulo**);**

GUI**.**Label **(**cajaLabel1**,**AppIUConstantes**.**MENSAJE\_AYUDA1**,**styleLabel**);**

GUI**.**Label **(**cajaLabel2**,**AppIUConstantes**.**MENSAJE\_AYUDA2**,**styleLabel**);**

ip1 **=** GUI**.**TextField**(**cajaTextField1**,** ip1**,** AppIUConstantes**.**MAX\_LENGTH**,**

styleTextField1**);**

GUI**.**Label **(**cajaLabelPunto1**,** AppIUConstantes**.**PUNTO**,**styleLabelPunto**);**

ip2**=** GUI**.**TextField**(**cajaTextField2**,** ip2**,** AppIUConstantes**.**MAX\_LENGTH**,**

styleTextField2**);**

GUI**.**Label **(**cajaLabelPunto2**,** AppIUConstantes**.**PUNTO**,**styleLabelPunto**);**

ip3**=** GUI**.**TextField**(**cajaTextField3**,** ip3**,** AppIUConstantes**.**MAX\_LENGTH**,**

styleTextField3**);**

GUI**.**Label **(**cajaLabelPunto3**,** AppIUConstantes**.**PUNTO**,**styleLabelPunto**);**

ip4 **=** GUI**.**TextField**(**cajaTextField4**,** ip4**,** AppIUConstantes**.**MAX\_LENGTH**,**s

tyleTextField4**);**

mensajeError**=this.**validarNumeroIngresados**(**ip1**,** ip2**,**ip3**,** ip4**);**

**if(**mensajeError**.**Equals**(**AppIUConstantes**.**CADENA\_VACIA**)){**

**if(**GUI**.**Button**(**cajaBotonAcepatar**,**AppIUConstantes**.**BOTON\_ACEPTAR**,**

styleBotonAceptar**)){**

seterListado**.**setHabilitarPantallaServidor**(false);**

**this.**AlmacenarIpEnMemoria**();**

**this.**contruirDireccionIp**();**

seterListado**.**cargarDatosBaseDatos**();**

**}**

**}else{**

GUI**.**Label **(**cajaMensajeError**,** mensajeError**,**styleMensajeError**);**

**}**

**}**

**}**

//Permite reajustar los componentes de interfaz segun la dimension de pantalla.

**public** void redimensionarPantalla**(**float width**,** float height **){**

**this.**ajustarTamanioLetra**();**

float alturaCajaTextField**=**52**\***height**/**100**;**

float alturaCajaLabelPunto**=**53**\***height**/**100**;**

Spinner **=** Resources**.**Load**(**"spinner\_XHigh"**)** **as** Texture**;**

cajaConfiguracionServidor**=new** Rect**(**width**/**10**,**height**/**10**,**

4**\***width**/**5**,**4**\***height**/**5**);**

cajaLoading**=new** Rect**(**width**/**2.0f **-** 35**\***height**/**200**,**

height**/**2.0f **-**35**\***height**/**200**,**35**\***height**/**100**,** 35**\***height**/**100**);**

cajaTitulo**=new** Rect**(**width**/**10**,**height**/**10**,**4**\***width**/**5**,**11**\***height**/**96**);**

cajaLabel1**=new** Rect**(**width**/**10**,**28**\***height**/**100**,**4**\***width**/**5**,**11**\***height**/**96**);**

cajaLabel2**=new** Rect**(**width**/**10**,**38**\***height**/**100**,**4**\***width**/**5**,**11**\***height**/**96**);**

cajaTextField1**=new** Rect**(**15**\***width**/**100**,**alturaCajaTextField**,**

16**\***width**/**100**,**15**\***height**/**100**);**

cajaTextField2**=new** Rect**(**17**\***width**/**100**+**16**\***width**/**100**,**alturaCajaTextField**,**

16**\***width**/**100**,**15**\***height**/**100**);**

cajaTextField3**=new** Rect**(**19**\***width**/**100**+**2**\***16**\***width**/**100**,**alturaCajaTextField**,**

16**\***width**/**100**,**15**\***height**/**100**);**

cajaTextField4**=new** Rect**(**85**\***width**/**100**-**16**\***width**/**100**,**alturaCajaTextField**,**

16**\***width**/**100**,**15**\***height**/**100**);**

cajaBotonAcepatar**=new** Rect**(**width**/**2**-**25**\***width**/**200**,**72**\***height**/**100**,**

25**\***width**/**100**,**15**\***height**/**100**);**

cajaMensajeError**=new** Rect**(**width**/**2**-**40**\***width**/**200**,**72**\***height**/**100**,**

40**\***width**/**100**,**10**\***height**/**100**);**

cajaLabelPunto1**=new** Rect**(**32**\***width**/**100**-**0.5f**\***width**/**100**,**alturaCajaLabelPunto**,**

5**\***width**/**100**,**5**\***height**/**100**);**

cajaLabelPunto2**=new** Rect**(**50**\***width**/**100**-**0.5f**\***width**/**100**,**alturaCajaLabelPunto**,**

5**\***width**/**100**,**5**\***height**/**100**);**

cajaLabelPunto3**=new** Rect**(**68**\***width**/**100**-**0.5f**\***width**/**100**,**alturaCajaLabelPunto**,**

5**\***width**/**100**,**5**\***height**/**100**);**

**}**

//Permite validar el ingreso de datos al sistema. Unicamente enteros.

**public** string validarNumeroIngresados**(**string ip1**,** string ip2**,** string ip3**,** string ip4**){**

int ip**;**

**try{**

ip**=**int**.**Parse**(**ip1**);**

**}catch(**Exception e**){**

**return** AppIUConstantes**.**MENSAJE\_ERROR**+**ip1**;**

**}**

**try{**

ip**=**int**.**Parse**(**ip2**);**

**}catch(**Exception e**){**

**return** AppIUConstantes**.**MENSAJE\_ERROR**+**ip2**;**

**}**

**try{**

ip**=**int**.**Parse**(**ip3**);**

**}catch(**Exception e**){**

**return** AppIUConstantes**.**MENSAJE\_ERROR**+**ip3**;**

**}**

**try{**

ip**=**int**.**Parse**(**ip4**);**

**}catch(**Exception e**){**

**return** AppIUConstantes**.**MENSAJE\_ERROR**+**ip4**;**

**}**

**return** AppIUConstantes**.**CADENA\_VACIA**;**

**}**

// Permite reajustar el tamaño de texto segun las dimensiones de pantalla

**public** void ajustarTamanioLetra**(){**

styleTextField1**.**fontSize**=(**int**)(**10**\***Screen**.**height**/**100**);**

styleTextField2**.**fontSize**=(**int**)(**10**\***Screen**.**height**/**100**);**

styleTextField3**.**fontSize**=(**int**)(**10**\***Screen**.**height**/**100**);**

styleTextField4**.**fontSize**=(**int**)(**10**\***Screen**.**height**/**100**);**

styleBotonAceptar**.**fontSize**=(**int**)(**8**\***Screen**.**height**/**100**);**

styleTitulo**.**fontSize**=(**int**)(**7**\***Screen**.**height**/**100**);**

styleLabel**.**fontSize**=(**int**)(**Screen**.**height**/**24**);**

styleMensajeError**.**fontSize**=(**int**)(**Screen**.**height**/**24**);**

styleLabelPunto**.**fontSize**=(**int**)(**10**\***Screen**.**height**/**100**);**

**}**

//Almacena la direccion ip en memoria.

**public** void AlmacenarIpEnMemoria**(){**

PlayerPrefs**.**SetInt**(**"ip1Almacenada"**,**int**.**Parse**(**ip1**));**

PlayerPrefs**.**SetInt**(**"ip2Almacenada"**,**int**.**Parse**(**ip2**));**

PlayerPrefs**.**SetInt**(**"ip3Almacenada"**,**int**.**Parse**(**ip3**));**

PlayerPrefs**.**SetInt**(**"ip4Almacenada"**,**int**.**Parse**(**ip4**));**

**}**

// Permite recuperar los valores almacenados de la direccion ip

**public** void recuperarValoresAlmacenados**(){**

ip1**=**AppIUConstantes**.**CADENA\_VACIA**+**PlayerPrefs**.**GetInt**(**"ip1Almacenada"**,**0**);**

ip2**=**AppIUConstantes**.**CADENA\_VACIA**+**PlayerPrefs**.**GetInt**(**"ip2Almacenada"**,**0**);**

ip3**=**AppIUConstantes**.**CADENA\_VACIA**+**PlayerPrefs**.**GetInt**(**"ip3Almacenada"**,**0**);**

ip4**=**AppIUConstantes**.**CADENA\_VACIA**+**PlayerPrefs**.**GetInt**(**"ip4Almacenada"**,**0**);**

**}**

//Construye la direecion ip con los datos ingresados desde la interfaz

**public** void contruirDireccionIp**(){**

int ip1**=**PlayerPrefs**.**GetInt**(**"ip1Almacenada"**,**0**);**

int ip2**=**PlayerPrefs**.**GetInt**(**"ip2Almacenada"**,**0**);**

int ip3**=**PlayerPrefs**.**GetInt**(**"ip3Almacenada"**,**0**);**

int ip4**=**PlayerPrefs**.**GetInt**(**"ip4Almacenada"**,**0**);**

ipServidor**=**ip1**+**AppIUConstantes**.**PUNTO**+**ip2**+**AppIUConstantes**.**PUNTO**+**ip3**+**

AppIUConstantes**.**PUNTO**+**ip4**;**

**}**

//Retorna el ip del servidor

**public** string getIpServidor**(){**

**return** **this.**ipServidor**;**

**}**

**}**

**Numero de Mesa:** **Permite traer de la base de datos información de la mesa y los carga en un objeto de tipo Mesa para su posterior uso.**

using UnityEngine;

using System.Collections;

using System.Text.RegularExpressions;

/\*Permite obtener una lista con objetos de tipo Mesa.

Son traidas de la base de datos del servidor\*/

public class NumeroMesa : MonoBehaviour {

private ArrayList list;

private AdministracionServidor administracionServidor;

private ArrayList listaMesa;

void Awake()

{

list=new ArrayList();

listaMesa=new ArrayList();

}

// Use this for initialization

void Start () {

administracionServidor=GetComponent<AdministracionServidor>();

}

// Update is called once per frame

void Update () {

}

//Permite cargar un ArrayLis con objetos de tipo mesa traidas de la base de datos

public IEnumerator getNumeroMesa(){

string cadena=AppIUConstantes.CADENA\_VACIA;

string urlServidor=AppIUConstantes.HTTP+administracionServidor.getIpServidor()+

AppIUConstantes.PHP\_SELECT\_MESA;

WWW palabra=new WWW(urlServidor);

yield return palabra;

if(palabra.error==null){

list.AddRange(Regex.Split(palabra.text,AppIUConstantes.FLECHA));

int indice=-1;

string idMesa;

string descripcionMesa;

for(int j=0;j<list.Count/2;j++){

indice++;

idMesa=list[indice]+AppIUConstantes.CADENA\_VACIA;

indice++;

descripcionMesa=list[indice]+AppIUConstantes.CADENA\_VACIA;

Mesa unaMesa=new Mesa(int.Parse(idMesa),descripcionMesa);

listaMesa.Add(unaMesa);

}

}

}

public ArrayList getListaMesa(){

return this.listaMesa;

}

}

**Redimensionar Pantalla: Este clase permite ubicar los botones en la cámara del dispositivo en función del tamaño de la pantalla. Estos botones son el Agregar Perdido, botón de Notificación y visualización del pedido y botones de Navegación. A la vez da las funcionalidades a estos botones.**

using UnityEngine;

using System.Collections;

using System;

/\*Este clase permite redimensionar los botones

segun el tamaño de la pantalla\*/

public class RedimencionarPantalla : MonoBehaviour {

public GUIStyle styleNotificacion;

public GUIStyle styleTitulo;

public GUIStyle styleBox;

public GUIStyle styleLabelMesa;

public GUIStyle styleComboBoxMesa;

public GUIStyle styleComboBoxSeleccion;

public GUIStyle styleBotonAceptar;

public GUIStyle styleEleccionNumeroMesa;

public GUIStyle styleMensajeError;

private Rect cajaNumeroMesa;

private Rect cajaTitulo;

private Rect cajaNotificacion;

private Rect cajaBotonAceptar;

private Rect cajaLabelMesa;

private Rect cajaComboBox;

private Rect cajaEleccionNumeroMesa;

private Rect cajaMensajeError;

private Texture circuloRojo;

public Texture[] texturaBotonAgregar;

public Texture[] texturaBotonNotificacion;

public Texture[] texturaBotonArriba;

public Texture[] texturaBotonAbajo;

public Texture[] texturaBotonSiguiente;

public Texture[] texturaBotonAnterior;

public Texture texturaTransparente;

public GUITexture imagenBotonAgregar;

public GUITexture imagenBotonNotificacion;

public GUITexture imagenBotonArriba;

public GUITexture imagenBotonAbajo;

public GUITexture imagenBotonSiguiente;

public GUITexture imagenBotonAnterior;

private bool botonAgregarPulsado;

private bool botonNotifiacionPulsado;

private bool botonArribaPulsado;

private bool botonAbajoPulsado;

private bool botonSiguientePulsado;

private bool botonAnteriorPulsado;

private bool habilitarPantallaDePedido;

private bool banderaEfectoBurbuja;

private bool banderaNumeroMesa;

private bool banderaMensajeError;

private bool editing;

private bool banderaMesa;

private bool habilitacionDiarioVideo;

private string[] items;

public string nombreSiguiente;

private string slectedItem;

private string eleccionMesa;

public Transform marcador;

public Transform ARCamara;

private int contadorPlatos;

private int eleccionNumeroMesa;

private Navegacion navegacion;

private SetearListado setearListado;

private VisualizadorModelo3D visualizadorModelo3D;

private AgregarMenuPedido botonAgregar;

private DefaultTrackableEventHandler defaultTrackableEventHandler;

private NumeroMesa numeroMesa;

private DibujarPantalla dibujarPantalla;

// Use this for initialization

void Awake()

{

botonAgregarPulsado=false;

botonNotifiacionPulsado=false;

botonArribaPulsado=false;

botonAbajoPulsado=false;

botonSiguientePulsado=false;

botonAnteriorPulsado=false;

habilitarPantallaDePedido=false;

banderaEfectoBurbuja=false;

banderaNumeroMesa=false;

editing = false;

banderaNumeroMesa=false;

banderaMesa=true;

contadorPlatos=0;

eleccionNumeroMesa=0;

slectedItem = AppIUConstantes.MESA\_SELECCION;

eleccionMesa=AppIUConstantes.MESA\_ELECCION\_MESA;

habilitacionDiarioVideo=false;

}

// Use this for initialization

void Start () {

dibujarPantalla=ARCamara.GetComponent<DibujarPantalla>();

defaultTrackableEventHandler=marcador.

GetComponent<DefaultTrackableEventHandler>();

navegacion= GetComponent<Navegacion>();

setearListado= GetComponent<SetearListado>();

botonAgregar=imagenBotonAgregar.GetComponent<AgregarMenuPedido>();

visualizadorModelo3D=GetComponent<VisualizadorModelo3D>();

numeroMesa=GetComponent<NumeroMesa>();

}

// Update is called once per frame

void Update () {

//-----------------Editor---------------

if(Input.GetKeyDown(KeyCode.A)){

setearListado.pasarCategoriaAnterior();

visualizadorModelo3D.cambiarObjetoAlCambiarCategoria();

}

if(Input.GetKeyDown(KeyCode.D)){

setearListado.pasarCategoriaSiguiente();

visualizadorModelo3D.cambiarObjetoAlCambiarCategoria();

}

if(Input.GetKeyDown(KeyCode.W)){

navegacion.reproducirAnimacionAnterior();

visualizadorModelo3D.visualizarAnteriorModelo3D();

}

if(Input.GetKeyDown(KeyCode.S)){

navegacion.reproducirAnimacionSiguiente();

visualizadorModelo3D.visualizarSiguienteModelo3D();

}

if(Input.GetKeyDown(KeyCode.Z)){

this.habilitarPantallaDePedido=true;

}

if(Input.GetKeyDown(KeyCode.X)){

if(dibujarPantalla.getBanderaIncializacionTea()){

ArrayList listaVacia=new ArrayList();

botonAgregar.setListaTeaMenuPedido(listaVacia);

dibujarPantalla.setBanderaIncializacionTea(false);

}

botonAgregar.AgregarMenuEnPantalla();

contadorPlatos++;

StartCoroutine(efectoBurbuja(Screen.width,Screen.height));;

dibujarPantalla.setBanderaHabilitacionBoton(true);

}

//----------------Android--------------------

if(!setearListado.getDescargandoDatos()&& !setearListado.

getDesactivarLoading()&& !setearListado.getHabilitarPantallaServidor()&&

!habilitarPantallaDePedido){

if(eleccionNumeroMesa==0){

if(banderaMesa){

StartCoroutine(numeroMesa.getNumeroMesa());

banderaMesa=false;

}

banderaNumeroMesa=true;

this.redimencionarInterfazMesa(Screen.width,Screen.height);

}else{

if(defaultTrackableEventHandler.getBanderaHabilitarPantallaDeNavegacion()){

this.redimencionarBotones(Screen.width,Screen.height);

this.seterValoresDefectoTextura();

if(Input.touchCount > 0){

for (var i = 0; i < Input.touchCount; ++i) {

//si realizo un touch en el boton agregar

if(imagenBotonAgregar.HitTest(Input.GetTouch(i).position)){

imagenBotonAgregar.guiTexture.texture=texturaBotonAgregar[1];

if(Input.GetTouch(i).phase==TouchPhase.Began){

botonAgregarPulsado = true;

}

if(Input.GetTouch(i).phase==TouchPhase.Ended){

imagenBotonAgregar.guiTexture.texture=texturaBotonAgregar[0];

}

//si realizo un touch en el boton Notificacion

} else if(imagenBotonNotificacion.HitTest(Input.GetTouch(i).position)){

imagenBotonNotificacion.guiTexture.texture=texturaBotonNotificacion[1];

if(Input.GetTouch(i).phase==TouchPhase.Ended){

botonNotifiacionPulsado = true;

imagenBotonNotificacion.guiTexture.texture=texturaBotonNotificacion[0];

}

}else if(imagenBotonArriba.HitTest(Input.GetTouch(i).position)){

imagenBotonArriba.guiTexture.texture=texturaBotonArriba[1];

if(Input.GetTouch(i).phase==TouchPhase.Began){

botonArribaPulsado = true;

}

if(Input.GetTouch(i).phase==TouchPhase.Ended){

imagenBotonArriba.guiTexture.texture=texturaBotonArriba[0];

}

} else if(imagenBotonAbajo.HitTest(Input.GetTouch(i).position)){

imagenBotonAbajo.guiTexture.texture=texturaBotonAbajo[1];

if(Input.GetTouch(i).phase==TouchPhase.Began){

botonAbajoPulsado = true;

}

if(Input.GetTouch(i).phase==TouchPhase.Ended){

imagenBotonAbajo.guiTexture.texture=texturaBotonAbajo[0];

}

}else if(imagenBotonSiguiente.HitTest(Input.GetTouch(i).position)){

imagenBotonSiguiente.guiTexture.texture=texturaBotonSiguiente[1];

if(Input.GetTouch(i).phase==TouchPhase.Began){

botonSiguientePulsado = true;

}

if(Input.GetTouch(i).phase==TouchPhase.Ended){

imagenBotonSiguiente.guiTexture.texture=texturaBotonSiguiente[0] ; }

} else if(imagenBotonAnterior.HitTest(Input.GetTouch(i).position)){

imagenBotonAnterior.guiTexture.texture=texturaBotonAnterior[1];

if(Input.GetTouch(i).phase==TouchPhase.Began){

botonAnteriorPulsado = true;

}

if(Input.GetTouch(i).phase==TouchPhase.Ended){

imagenBotonAnterior.guiTexture.texture=texturaBotonAnterior[0];

}

}

}

}

} else{

if(contadorPlatos==0){

imagenBotonNotificacion.guiTexture.texture=texturaTransparente;

imagenBotonNotificacion.transform.position = Vector3.zero;

imagenBotonNotificacion.transform.localScale = Vector3.zero;

}else{

if(Input.touchCount > 0){

for (var i = 0; i < Input.touchCount; ++i) {

if(imagenBotonNotificacion.HitTest(Input.GetTouch(i).position)){

imagenBotonNotificacion.guiTexture.texture=texturaBotonNotificacion[1];

if(Input.GetTouch(i).phase==TouchPhase.Ended){

botonNotifiacionPulsado = true;

imagenBotonNotificacion.guiTexture.texture=texturaBotonNotificacion[0];

}

}

}

}

}

this.borrarGUITexture();

}

}

}

if(botonAgregarPulsado){

if(dibujarPantalla.getBanderaIncializacionTea()){

ArrayList listaVacia=new ArrayList();

botonAgregar.setListaTeaMenuPedido(listaVacia);

dibujarPantalla.setBanderaIncializacionTea(false);

}

botonAgregar.AgregarMenuEnPantalla();

contadorPlatos++;

StartCoroutine(efectoBurbuja(Screen.width,Screen.height));

dibujarPantalla.setBanderaHabilitacionBoton(true);

}else

if(botonNotifiacionPulsado){

this.habilitarPantallaDePedido=true;

}

//Permite cambiar el modelo 3d y reproducir animacion de las letras.

if(botonArribaPulsado){

navegacion.reproducirAnimacionAnterior();

visualizadorModelo3D.visualizarAnteriorModelo3D();

}else if(botonAbajoPulsado){

navegacion.reproducirAnimacionSiguiente();

visualizadorModelo3D.visualizarSiguienteModelo3D();

}

//Permite cambiar de Categoria

if(botonSiguientePulsado){

setearListado.pasarCategoriaSiguiente();

visualizadorModelo3D.cambiarObjetoAlCambiarCategoria();

}else if(botonAnteriorPulsado){

setearListado.pasarCategoriaAnterior();

visualizadorModelo3D.cambiarObjetoAlCambiarCategoria();

}

//Las variables logicas vuelven a setearse en false para un uso posterior

botonAgregarPulsado = false;

botonNotifiacionPulsado=false;

botonArribaPulsado=false;

botonAbajoPulsado=false;

botonSiguientePulsado=false;

botonAnteriorPulsado=false;

}

/\*Permite dibujar interfaz de mesa\*/

void OnGUI() {

if(banderaNumeroMesa){

GUI.Box(cajaNumeroMesa,AppIUConstantes.CADENA\_VACIA, styleBox);

GUI.Label (cajaTitulo, AppIUConstantes.MESA\_TITULO,styleTitulo);

GUI.Label (cajaLabelMesa, AppIUConstantes.MESA\_LABEL,styleLabelMesa);

GUI.Label (cajaEleccionNumeroMesa, eleccionMesa,styleEleccionNumeroMesa);

if (GUI.Button(cajaComboBox, slectedItem,styleComboBoxMesa))

{

editing = true;

}

if (editing)

{

for (int x = 0; x < numeroMesa.getListaMesa().Count; x++)

{

if (GUI.Button(new Rect(cajaComboBox.x, (cajaComboBox.height \* x) +

cajaComboBox.y + cajaComboBox.height, cajaComboBox.width,

cajaComboBox.height), ((Mesa)numeroMesa.getListaMesa()[x]).

getIdMesa()+"",styleComboBoxSeleccion))

{

slectedItem = ((Mesa)numeroMesa.getListaMesa()[x]).getIdMesa()+"";

editing = false;

eleccionMesa=slectedItem;

banderaMensajeError=false;

}

}

}

if (GUI.Button(cajaBotonAceptar, AppIUConstantes.BOTON\_ACEPTAR,styleBotonAceptar))

{

GUI.Label (cajaMensajeError, AppIUConstantes.MESAJE\_ERROR\_COMBOBOX,

styleMensajeError);

try{

habilitacionDiarioVideo=true;

eleccionNumeroMesa=int.Parse(slectedItem);

}catch(Exception e){

banderaMensajeError=true;

habilitacionDiarioVideo=false;

}

banderaNumeroMesa=false;

}

if(banderaMensajeError){

GUI.Label (cajaMensajeError, AppIUConstantes.MESAJE\_ERROR\_COMBOBOX,

styleMensajeError);

}

}else{

if(contadorPlatos>0){

this.redimencionarBotones(Screen.width, Screen.height);

imagenBotonNotificacion.guiTexture.texture=texturaBotonNotificacion[0];

GUI.Label (cajaNotificacion,contadorPlatos+AppIUConstantes.CADENA\_VACIA,

styleNotificacion);

}

}

}

/\*Permite redimencionar los componentes de interfaz de mesa\*/

public void redimencionarInterfazMesa(float ancho, float alto){

cajaNumeroMesa=new Rect(20\*ancho/100,20\*alto/100,

60\*ancho/100,60\*alto/100);

cajaTitulo=new Rect(20\*ancho/100,20\*alto/100,

60\*ancho/100,8\*alto/100);

cajaComboBox=new Rect(100\*ancho/100-22\*ancho/100-30\*ancho/100,

37\*alto/100,30\*ancho/100,10\*alto/100);

cajaLabelMesa=new Rect(27\*ancho/100,38\*alto/100,

30\*ancho/100,8\*alto/100);

cajaBotonAceptar=new Rect(22\*ancho/100,55\*alto/100,

25\*ancho/100,15\*alto/100);

cajaEleccionNumeroMesa=new Rect(62\*ancho/100,57\*alto/100,

25\*ancho/100,15\*alto/100);

cajaMensajeError=new Rect(22\*ancho/100,70\*alto/100,

25\*ancho/100,15\*alto/100);

styleTitulo.fontSize=(int)(7\*alto/100);

styleLabelMesa.fontSize=(int)(6\*alto/100);

styleComboBoxSeleccion.fontSize=(int)(6\*alto/100);

styleComboBoxMesa.fontSize=(int)(6\*alto/100);

styleBotonAceptar.fontSize=(int)(8\*Screen.height/100);

styleEleccionNumeroMesa.fontSize=(int)(6\*alto/100);

styleMensajeError.fontSize=(int)(6\*alto/100);

}

//Este metodo permite posicionar en funcion del tamaño de la pantalla los diferentes botones

public void redimencionarBotones(float ancho, float alto){

if(banderaEfectoBurbuja){

styleNotificacion.fontSize=(int)(0.1f\*Screen.height/100);

cajaNotificacion=new Rect(12\*ancho/100,3\*alto/100,4\*ancho/100,4\*ancho/100);

}else{

styleNotificacion.fontSize=(int)(0.1f\*Screen.height/100);

cajaNotificacion=new Rect(12\*ancho/100,3\*alto/100,3\*ancho/100,3\*ancho/100);

}

//Boton Agregar

imagenBotonAgregar.transform.position = Vector3.zero;

imagenBotonAgregar.transform.localScale = Vector3.zero;

imagenBotonAgregar.guiTexture.pixelInset=new Rect(ancho\*86/100,

alto\*77/100,13\*ancho/100,13\*ancho/100);

//Boton Notificacion

imagenBotonNotificacion.transform.position = Vector3.zero;

imagenBotonNotificacion.transform.localScale = Vector3.zero;

imagenBotonNotificacion.guiTexture.pixelInset=new Rect(ancho\*1/100,

alto\*79/100,13\*ancho/100,13\*ancho/100);

//Boton Anterior

imagenBotonAnterior.transform.position = Vector3.zero;

imagenBotonAnterior.transform.localScale = Vector3.zero;

imagenBotonAnterior.guiTexture.pixelInset=new Rect(ancho\*100/100-ancho\*27/100-10\*ancho/100,

alto\*1/100,10\*ancho/100,10\*ancho/100);

//Boton Siguiente

imagenBotonSiguiente.transform.position = Vector3.zero;

imagenBotonSiguiente.transform.localScale = Vector3.zero;

imagenBotonSiguiente.guiTexture.pixelInset=new Rect(ancho\*100/100-ancho\*1/100-10\*ancho/100,

alto\*1/100,10\*ancho/100,10\*ancho/100);

//Boton Arriba

imagenBotonArriba.transform.position = Vector3.zero;

imagenBotonArriba.transform.localScale = Vector3.zero;

imagenBotonArriba.guiTexture.pixelInset=new Rect(ancho\*76/100,

alto\*22/100,10\*ancho/100,10\*ancho/100);

//Boton Abajo

imagenBotonAbajo.transform.position = Vector3.zero;

imagenBotonAbajo.transform.localScale = Vector3.zero;

imagenBotonAbajo.guiTexture.pixelInset=new Rect(ancho\*76/100,

alto\*1/100,10\*ancho/100,10\*ancho/100);

}

// Permite cambiar de tamaño el circulo de color rojo

// de notificacion.

public IEnumerator efectoBurbuja(float ancho, float alto){

banderaEfectoBurbuja=true;

yield return new WaitForSeconds(AppIUConstantes.TIEMPO\_ESPERA1);

banderaEfectoBurbuja=false;

}

//Permite dejar por defecto los valores de textura del boton

public void seterValoresDefectoTextura(){

if(Input.touchCount==0){

imagenBotonAgregar.guiTexture.texture=texturaBotonAgregar[0];

imagenBotonNotificacion.guiTexture.texture=texturaBotonNotificacion[0];

imagenBotonArriba.guiTexture.texture=texturaBotonArriba[0];

imagenBotonAbajo.guiTexture.texture=texturaBotonAbajo[0];

imagenBotonSiguiente.guiTexture.texture=texturaBotonSiguiente[0];

imagenBotonAnterior.guiTexture.texture=texturaBotonAnterior[0];

}

}

//Hace desaparecer los botones cuando el marcador no es detectado.

public void borrarGUITexture(){

imagenBotonAgregar.guiTexture.texture=texturaTransparente;

imagenBotonArriba.guiTexture.texture=texturaTransparente;

imagenBotonAbajo.guiTexture.texture=texturaTransparente;

imagenBotonSiguiente.guiTexture.texture=texturaTransparente;

imagenBotonAnterior.guiTexture.texture=texturaTransparente;

imagenBotonAgregar.transform.position = Vector3.zero;

imagenBotonAgregar.transform.localScale = Vector3.zero;

imagenBotonAnterior.transform.position = Vector3.zero;

imagenBotonAnterior.transform.localScale = Vector3.zero;

imagenBotonSiguiente.transform.position = Vector3.zero;

imagenBotonSiguiente.transform.localScale = Vector3.zero;

imagenBotonArriba.transform.position = Vector3.zero;

imagenBotonArriba.transform.localScale = Vector3.zero;

imagenBotonAbajo.transform.position = Vector3.zero;

imagenBotonAbajo.transform.localScale = Vector3.zero;

}

//Getter y Setter.

public void setHabilitarPantallaDePedido(bool habilitarPantallaDePedido ){

this.habilitarPantallaDePedido=habilitarPantallaDePedido;

}

public bool getHabilitarPantallaDePedido(){

return this.habilitarPantallaDePedido;

}

public int getContadorPlato(){

return contadorPlatos;

}

public void descremenarContadorPlatos(){

contadorPlatos--;

}

public void inicializarContadorPlatos(){

contadorPlatos=0;

}

public int getEleccionNumeroMesa(){

return this.eleccionNumeroMesa;

}

public bool getHabilitacionDiarioVideo(){

return this.habilitacionDiarioVideo;

}

}

Pruebas

Esta etapa se realizó la verificación y validación del prototipo para mostrar que cumple su especificación y las funciones que provee no presentan errores ni defectos. En caso de existir anomalías se procede a realizar las correcciones pertinentes y se somete al proceso de prueba nuevamente.

Casos de pruebas

Estas pruebas fueron escritas con el fin de ayudar a la implementación del prototipo y verificar si cumplen los requisitos funcionales y no funcionales del sistema. Los casos de pruebas debe ser ejecutadas en orden, para ello se debe iniciar el sistema e ingresar los datos de entrada con el fin de verificar las salidas esperadas. Los casos de prueba escritos, incluyen una descripción de la funcionalidad que se va a probar, la cuál es tomada de los [casos de us](http://es.wikipedia.org/wiki/Caso_de_uso)os antes desarrollados. La notación para los casos de prueba es CPXX(N/A), donde CP: Caso de Prueba, XX es el número de caso de prueba y (N/A) es el flujo normal y alternativo. Estos casos de prueba deben estar en orden.

# Alta de Pedido- Sistema de Atención Aumentado.

# CP01N: Flujo Normal

|  |  |
| --- | --- |
| **Propósito** | Realizar un alta de un pedido y almacenarlo en la base de datos. |
| **Actor** | Usuario Cliente |
| **Pre-requisitos** | Se debe generar la carta gourmet con realidad aumentada. |
| **Datos de Prueba** | * Nombre Plato: Pizza Especial; Precio: 60$ * Nombre Plato: Hamburguesa Completa; Precio: 37$ |
| **Pasos** | 1. Selecciona el menú con el nombre “Pizza Especial” 2. Presiona el botón “Agregar” 3. Selecciona el menú con el nombre “Hamburguesa Completa” 4. Presiona el botón “Agregar” 5. Presionar el botón “Ver Pedidos” 6. Presiona el botón “Enviar” |
| **Resultado esperado** | Los resultados pueden verse en el Sistema de Gestión de Pedidos  Resultado en la Tabla PEDIDO   * PEDIDO\_ID: 1; PEDIDO\_FECHA: Fecha de Sistema; PEDIDO\_TOTAL: 97$, MESA\_ID:1   Resultado en la Tabla DETALLE\_PEDIDO:   * DET\_PED\_ID: 1; DET\_PED \_CANTIDAD: 1; DET\_PED \_ESTADO: REALIZADO; PEDIDO\_ID: 1; MENU\_ID: 1. * DET\_PED \_ID: 2; DET\_PED \_CANTIDAD: 1; DET\_PED \_ESTADO: REALIZADO; PEDIDO\_ID: 1; MENU\_ID: 4. |

# Consultar Pedido- Sistema de Atención Aumentado.

# CP02N: Flujo Normal

|  |  |
| --- | --- |
| **Propósito** | Consular los pedidos realizados. |
| **Actor** | Usuario Cliente |
| **Pre-requisitos** | Se debe ejecutar el caso de prueba CP01N. (Alta de Pedido flujo normal) |
| **Datos de Prueba** | Ninguno |
| **Pasos** | 1. Presiona el botón “Ver Pedido” |
| **Resultado esperado** | Pantalla de Pedido   * Pizza Especial; Precio: 60$, Cantidad:1; Estado: Enviado * Hamburguesa Completa; Precio: 37$; Cantidad:1; Estado: Enviado   TOTAL: 97$ |

# Modificar Pedido- Sistema de Atención Aumentado.

# CP03N: Flujo Normal

|  |  |
| --- | --- |
| **Propósito** | Modificar un Pedido antes de ser almacenado en la base de datos. |
| **Actor** | Usuario Cliente |
| **Pre-requisitos** | Se debe ejecutar el caso de prueba CP01N nuevamente, pero sin realizar el paso 6. (Alta de Pedido flujo normal) |
| **Datos de Prueba** | Pantalla de Pedido   * Nombre Plato: Pizza Especial; Precio: 60$; Cantidad:1; Estado: No Enviado * Nombre Plato: Hamburguesa Completa, Precio: 37$, Cantidad:1; No Enviado   TOTAL: 97$ |
| **Pasos** | 1. Presionar el botón “Ver Pedido” 2. Presiona el botón para Eliminar el plato con el nombre “Hamburguesa Completa” |
| **Resultado esperado** | La Pantalla de Pedido se actualiza:   * Nombre Plato: Pizza Especial; Precio: 60$; Cantidad:1; Estado: No Enviado   TOTAL: 60$ |

# Login del Sistema-Sistema de Gestión de Pedido.

### CP04N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | Realizar el Login exitoso para un usuario administrador. |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Debe existir al menos un usuario registrado. |
| **Datos de Prueba** | User=pchavez  Password=chavez123 |
| **Pasos** | 1. Inicia el sistema 2. Ingresa datos 3. Presiona el botón “Ingresar” |
| **Resultado esperado** | Redirección a Pantalla Pedido. |

### CP05A: Flujo alternativo

|  |  |
| --- | --- |
| **Propósito** | Realizar el Login fallido con un usuario administrador incorrecto |
|  |  |
| **Pre-requisitos** | Debe existir al menos un usuario administrador en la base de datos. |
| **Datos de Prueba** | Usuario Incorrecto  User=pchaves  Password=chavez123  Contraseña Incorrecta  User=pchavez  Password=chaves123 |
| **Pasos** | 1. Inicia el sistema 2. Ingresa datos 3. Presiona el botón “Ingresar” |
| **Resultado esperado** | Visualiza pantalla de error con la leyenda “El usuario o contraseña es incorrecto”. |

# Agregar Diario-Sistema de Gestión de Pedido.

# CP06N-Flujo Normal.

|  |  |
| --- | --- |
| **Propósito** | Agregar enlace de diario digital a la base de datos. |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Se debe ejecutar el caso de prueba CP04N. (Login del sistema flujo normal) |
| **Datos de Prueba** | Nombre: El Tribuno  URL: http://www.eltribuno.info/ |
| **Pasos** | 1. Presionar la pestaña “Diario” 2. Presionar el botón “Agregar” 3. Cargar los datos de prueba. 4. Presionar el botón “Aceptar” |
| **Resultado esperado** | El enlace de Diario digital se cargar en la tabla y en la base de datos del sistema. |

# Consultar Diarios Digitales- Sistema de Atención Aumentado.

# CP07N: Flujo Normal

|  |  |
| --- | --- |
| **Propósito** | Consular los diarios digitales más leídos. |
| **Actor** | Usuario Cliente |
| **Pre-requisitos** | Se tiene que enfocar a la carta gourmet para generar el enlace aumentado. Se debe ejecutar previamente el caso de prueba CP06N (Agregar Diario). |
| **Datos de Prueba** | Ninguno |
| **Pasos** | 1. Presiona el enlace aumentado. 2. Presionar le botón “El tribuno” |
| **Resultado esperado** | Se re direcciona a la pagina del tribuno de Jujuy. |

# Agregar Video- Sistema de Gestión de Pedido

### CPO8N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | Almacena un enlace de video en la base de datos del sistema |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Se debe ejecutar el caso de prueba CP04N (Login del sistema flujo Normal) |
| **Datos de Prueba** | Nombre: Pizza Especial  URL: http://www.youtube.com/watch?v=Nr\_FD3kWrsw |
| **Pasos** | 1. Presionar la pestaña “Video” 2. Presionar el botón “Agregar” 3. Cargar datos de prueba 4. Presionar el botón “Aceptar” |
| **Resultado esperado** | Se almacena el enlace de video en la base de datos. |

# Cargar Video- Sistema de Gestión de Pedido

### CPO9N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | Carga un video para que el sistema de atención aumentado pueda reproducirlo. |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Debe haber por lo menos un enlace almacenado en la base de datos. Se debe ejecutar el caso de prueba CP08N con anterioridad (Agregar Video). |
| **Datos de Prueba** | Nombre: Pizza Especial  URL: http://www.youtube.com/watch?v=Nr\_FD3kWrsw |
| **Pasos** | 1. Presionar la pestaña “Video” 2. Seleccione el video con el nombre Pizza Especial. 3. Presione el botón “Cargar Video” |
| **Resultado esperado** | La fila seleccionada se cambia de color indicando un estado ACTIVO. |

# Consultar Videos- Sistema de Atención Aumentado.

### CP10N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | Consultar Videos de recomendaciones de chef. |
| **Actor** | Usuario Cliente |
| **Pre-requisitos** | Se tiene que enfocar a la carta gourmet para generar el enlace aumentado. Se debe ejecutar con anterioridad el CP08N (Agregar Video). |
| **Datos de Prueba** | Ninguno |
| **Pasos** | 1. Presiona el enlace aumentado. 2. Presionar le botón “Recomendación del Chef” |
| **Resultado esperado** | Se re direcciona a la página de youtube para cargar el video con el nombre Recomendación del Chef. |

# Navegar por la Carta Gourmet- Sistema de Atención Aumentado.

### CPO11N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | Navegar por la Carta gourmet |
|  | Usuario Cliente |
| **Pre-requisitos** | Se tiene que enfocar la cámara a la carta gourmet para generar la carta aumentada. |
| **Datos de Prueba** | Ninguno |
| **Pasos** | 1. Presiona el botón “siguiente” de Categoría 2. El sistema cambia de Categoría de PLATO a BEBIDA. 3. Presiona el botón “siguiente” de Menú tres veces |
| **Resultado esperado** | El sistema desplaza el selector de menú en la posición con el nombre CAPUCHINO. |

# Consulta Pedido-Sistema de Gestión de Pedido

### CP12N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | El administrador consulta el pedido almacenado en la base de datos. |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Debe existir al menos un pedido guardado. Se debe ejecutar el caso de prueba CP01N (Alta de Pedido) |
| **Datos de Prueba** | Pantalla de Pedido  Id: 1, Fecha: Fecha del sistema, total: 97$, Mesa: 1. |
| **Pasos** | 1. Presionar la pestaña “Pedido” 2. Selecciona el pedido con id: 1. 3. Presiona el botón “Detalle” |
| **Resultado esperado** | Visualiza el detalle de esa consulta:   * Id de Menú=1, Nombre del Plato: PIZZA ESPECIAL, Precio: 60, Cantidad:1, Estado: Enviado * Id de Menú=4, Nombre del Plato: HAMBURQUESA ESPECIAL, Precio: 37, Cantidad:1, Estado: Enviado |

### CPO13A: Flujo Alternativo

|  |  |
| --- | --- |
| **Propósito** | El administrador consulta el pedido almacenado ingresando un intervalo de fechas. |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Debe existir al menos un pedido guardado. Se debe ejecutar el caso de prueba CP01N (Alta de Pedido) |
| **Datos de Prueba** | Fecha Inicial: Fecha del sistema -4 meses.  Fecha Final: Fecha del sistema |
| **Pasos** | 1. Presionar la pestaña “Pedido” 2. Ingreso fecha inicial y final. 3. Selecciono el pedido con id: 1 4. Presiono el botón detalle |
| **Resultado esperado** | Visualiza el detalle de esa consulta:   * Id de Menú=1, Nombre del Plato: PIZZA ESPECIAL, Precio: 60, Cantidad:1, Estado: Enviado * Id de Menú=4, Nombre del Plato: HAMBURQUESA ESPECIAL, Precio: 37, Cantidad:1, Estado: Enviado |

# Generar Pedido-Sistema de Gestión de Pedido

### CP14N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | Generar un pdf de una Consulta de Pedido. |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Se debe realizar el caso de prueba CP12N O CP12A (Consultar Pedido) |
| **Datos de Prueba** | Visualiza el detalle de esa consulta:   * Id de Menú=1, Nombre del Plato: PIZZA ESPECIAL, Precio: 60, Cantidad:1, Estado: Enviado * Id de Menú=4, Nombre del Plato: HAMBURQUESA ESPECIAL, Precio: 37, Cantidad:1, Estado: Enviado |
| **Pasos** | 1. Presionar el botón “imprimir” |
| **Resultado esperado** | Se genera archivo en formato pdf que se guarda en Mis Documentos. |

# Eliminar Video-Sistema de Gestión de Pedido

### CP15N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | Eliminar un enlace de video de la base de datos. |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Debe existir al menos un video en la base de datos. Se debe ejecutar el caso de prueba CP08N previamente.(Agregar video) |
| **Datos de Prueba** | Nombre: Pizza Especial  URL: http://www.youtube.com/watch?v=Nr\_FD3kWrsw |
| **Pasos** | 1. Presionar la pestaña “Video” 2. Seleccione el video con el nombre Pizza Especial. 3. Presione el botón “Eliminar” |
| **Resultado esperado** | El enlace de video se elimina de la base de datos. |

# Eliminar Diario--Sistema de Gestión de Pedido

### CP16N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | Eliminar enlace de diario digital a la base de datos. |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Debe haber por lo menos un enlace cargado en el sistema. Se debe ejecutar el caso de prueba CP06N previamente(Agregar Diario) |
| **Datos de Prueba** | Nombre: El Tribuno  URL: http://www.eltribuno.info/ |
| **Pasos** | 1. Presionar la pestaña “Diario” 2. Seleccionar el enlace de diario digital con el nombre “El Tribuno” 3. Presionar el botón “Eliminar” |
| **Resultado esperado** | El enlace de diario digital se elimina de la base de datos, |

# Modificar Precio-Sistema de Gestión de Pedido

### CPO17N: Flujo normal

|  |  |
| --- | --- |
| **Propósito** | Modifica el precio de un plato y guarda los cambios en la base de datos |
| **Actor** | Usuario Administrador |
| **Pre-requisitos** | Se debe ejecutar el caso de prueba CP06N (Login del sistema flujo normal) |
| **Datos de Prueba** | Id:1  Nombre Plato: Pizza Especial  Precio: 65  Descripción: Descripción  Categoría: PLATOS |
| **Pasos** | 1. Presionar la pestaña “Menú” 2. Seleccionar el menú que se va a modificar 3. Presionar el botón “Modificar Precio” 4. Cargar los datos de prueba 5. Presionar el botón “Aceptar”. |
| **Resultado esperado** | El cambio realizado se guarda en la base de datos. |