**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, lo anota o toma el pedido verbalmente. Después me pasa el pedido, 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 agilizar 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 las compras de insumos como las bebidas, los materiales para los platos y 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, mi 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. Sí, se cuenta con una PC y un dispositivo móvil, pero me gustaría comprar una Tablet.
13. Si, se 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 con Realidad Aumentada 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 elegir una mesa, 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. Además el administrador podrá configurar el acceso al servidor mediante el ingreso de una dirección IP. Esta dirección IP debe ser ingresada al sistema móvil.

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.
* Debe permitir a los clientes poder elegir una mesa.
* 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.
* Además el administrador podrá configurar el acceso al servidor mediante el ingreso de una dirección IP

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

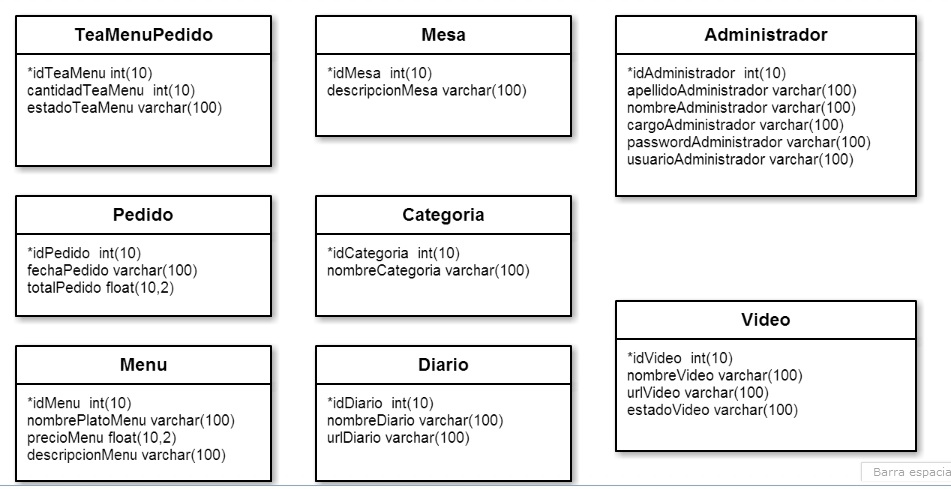


Diagrama Entidad - Relacion Global

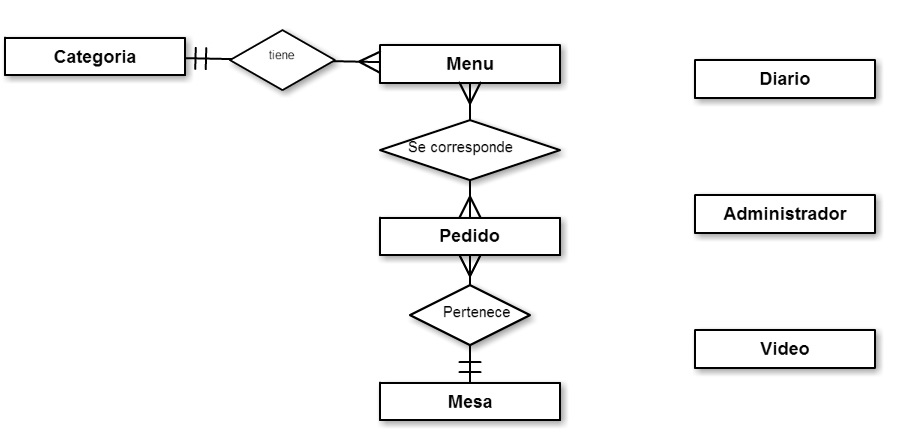


Diagrama de Martin

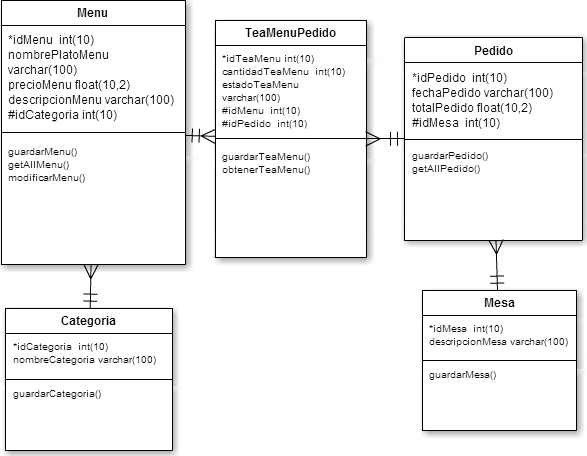
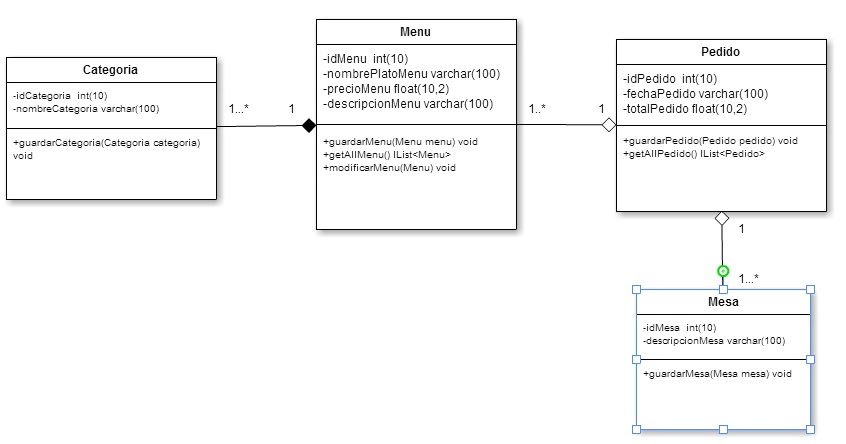
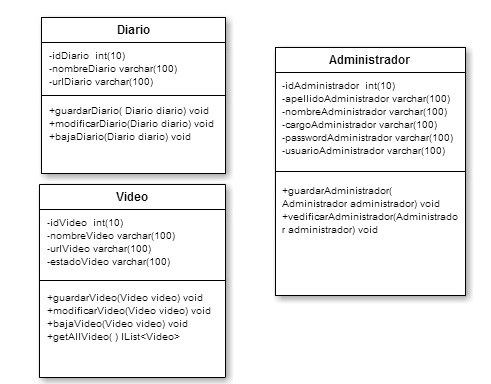


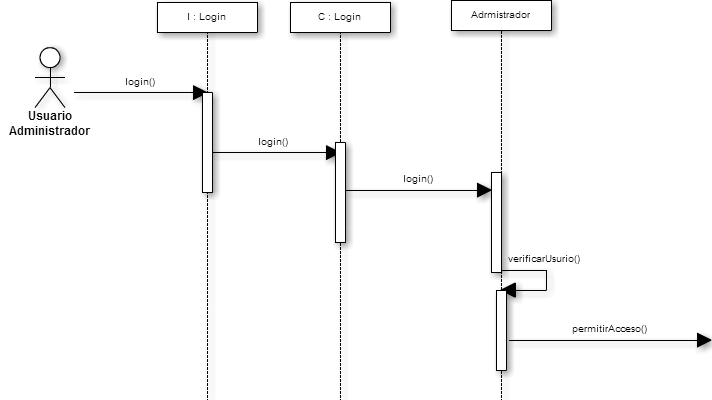
Diagrama de clases



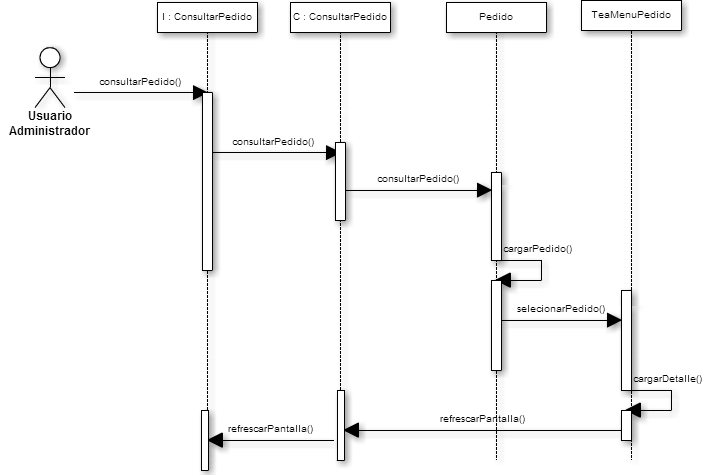


Diagramas de secuencia

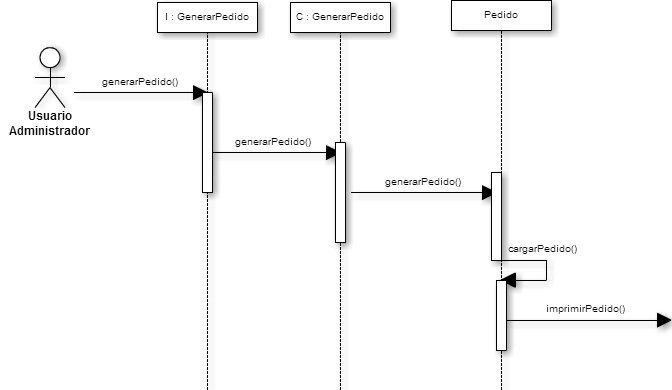
Login



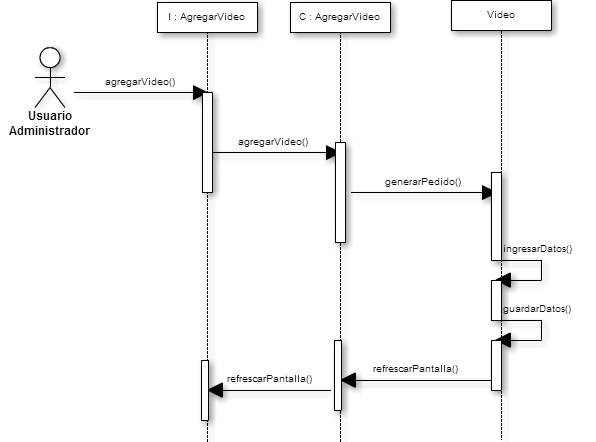
Consultar Pedido



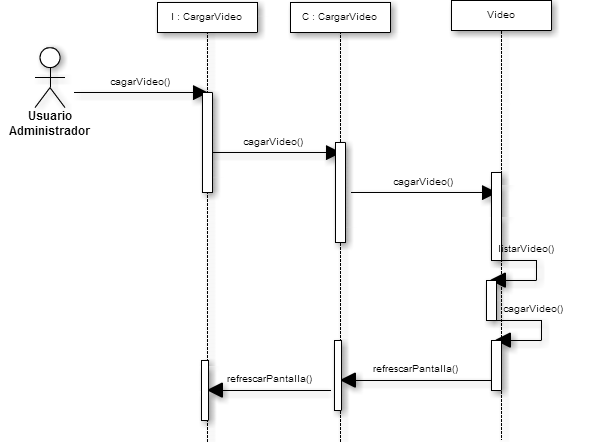
Generar Pedido



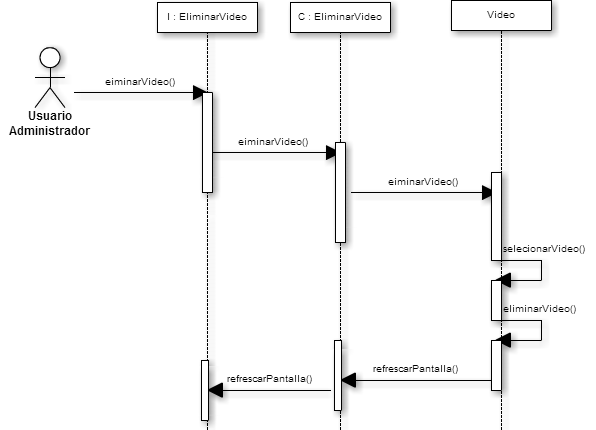
Agregar Video



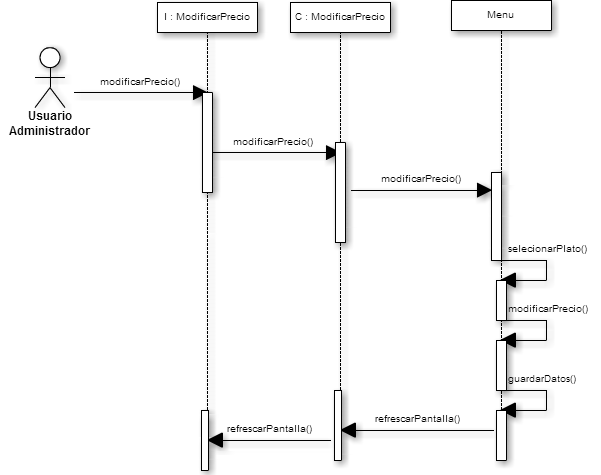
Cargar Video



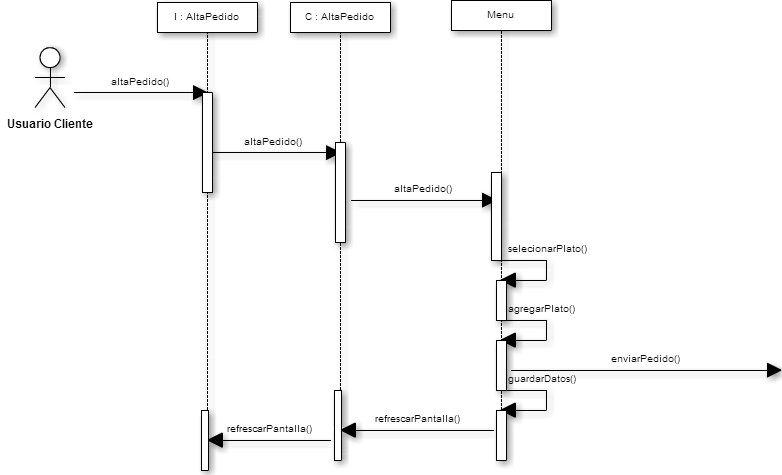
Eliminar Video



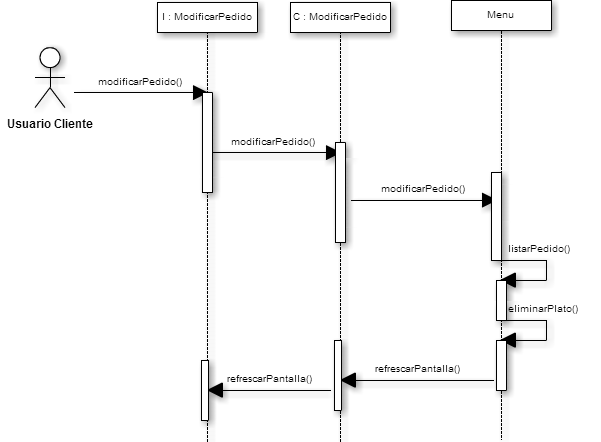
Modificar Precio



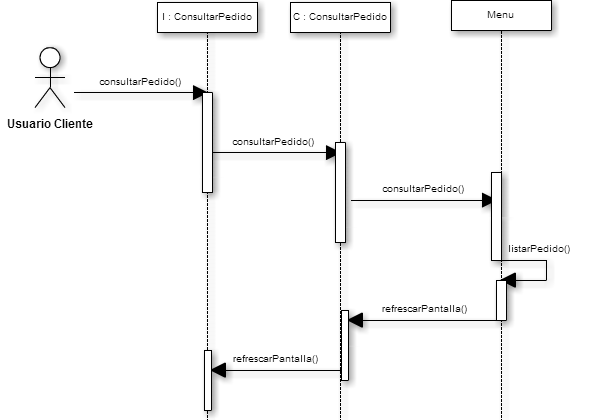
Alta de pedido



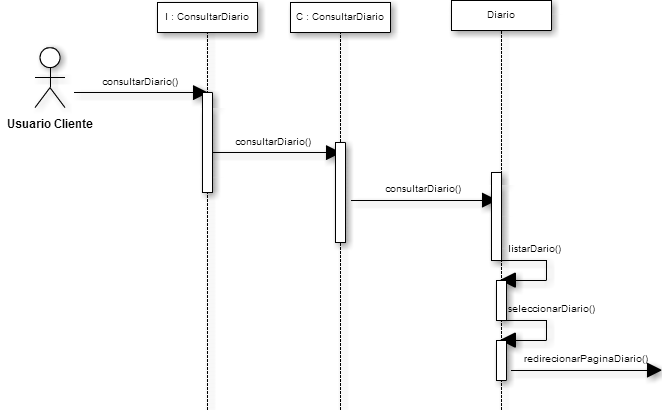
Modificar Pedido



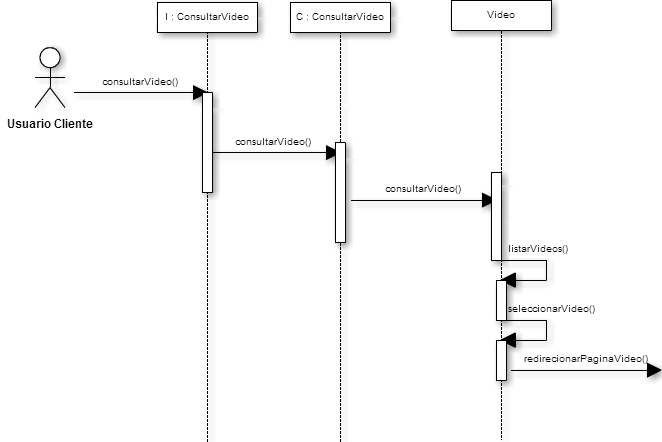
Consultar Pedido



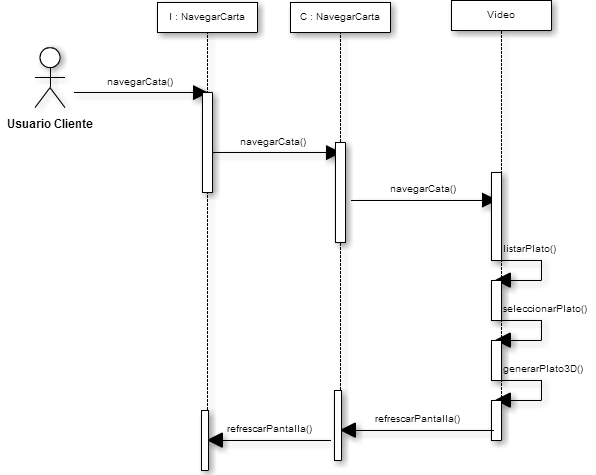
Consultar Diarios Digitales



Consultar Video



Navegar Carta Gourmet



Definición de Interfaz de usuario

**Interfaz de usuario**

Sistema de Atención Aumentada: Es el sistema que manejan los clientes del negocio para poder realizar sus pedidos a través de un dispositivo móvil.

Al ejecutar la aplicación móvil, los usuarios del prototipo del sistema se encuentran con la interfaz de presentación de la figura x.1



Figura x.1: Splash de la aplicación

Luego del Splash de la aplicación, se carga la Interfaz de selección de mesa.

***Interfaz de selección de mesa***: Al inicio de la aplicación se solicita al usuario agregar el numero de mesa que ocupa, para asegurar la correspondencia con el pedido que realizara. Si el numero no es ingresado la aplicación no continuara a la pantalla de navegación.



Figura x. : Pantalla de navegación.

***Interfaz de navegación****:* permite visualizar los platos que el usuario puede agregar al pedido digital, para luego consumir.



Figura x. : Pantalla de navegación.

En la carta gourmet puede verse en la parte izquierda las listas de platos que el cliente puede pedir. Mientras en la parte derecha se observa una representación 3D del plato, correspondiente al plato seleccionado en la lista antes mencionada.

En la pantalla de la aplicación se puede observar 3 tipos de botones principales:

* Botones de navegación: permite al cliente navegar por la carta gourmet. Podemos observar dos grupos de botones, en la parte inferior izquierda se encuentran los botones para cambiar de categoría de platos.

Mientras que en la parte inferior derecha se encuentran lo botones para cambiar de platos de una misma categoría.

* Botón Agregar: El botón se encuentra ubicado en la parte superior derecha de la pantalla. Permite agregar el plato actual a la lista de pedido. Si el boton es pulsado n veces en un mismo plato, se agregara en plato con una cantidad igual a n.
* Botón Menú Pedidos: Permite al usuario ingresar a la Interfaz de Pedido. El botón se encuentra ubicado a en la parte superior izquierda de la pantalla.

También se puede observar en la carta gourmet 2 elementos opcionales:

* Diarios Digitales: Permite al usuario tener acceso a los diarios digitales más populares. Se encuentra ubicado en la parte inferior de la carta gourmet. Al pulsar sobre el elemento se muestra un listado de los diarios a los cuales el usuario puede acceder. Cada botón de la lista redirecciona a la página digital del diario seleccionado.



Figura x. : Pantalla de Diarios Digitales.

* Video de entretenimiento: Permite al usuario tener acceso al video recomendado por el administrador. Se encuentra ubicado en la parte superior derecha de la carta gourmet. Al pulsar sobre el elemento se redirecciona a la página de youtube del video.



Figura x. : Video de entretenimiento.

***Interfaz de Pedido:*** Muestra los platos y bebidas que el usuario agrego a su pedido.

La interfaz lista todos los elementos que fueron agregados al pedidos, con su cantidad y precio correspondiente.

Además en la parte derecha de la lista se puede observar una "X" que permite eliminar un elemento del pedido.



Figura x. : Pantalla de pedido.

Cuando el cliente a completado su pedido debe presionar el Botón Enviar para que sea recibido por los dueños del negocio. En caso contrario, debe presionar el Botón Cancelar para volver a la pantalla de navegación y seguir agregando elementos a su pedido.

***Interfaz de configuración de servidor:*** Permite al usuario configurar la conexión del dispositivo móvil con el servidor.

En el caso de errores con la conexión al servidor, el dispositivo movil genera en la pantalla un cuadro para colocar en forma manual la ip del servidor.



Figura x. : Error al conectar servidor.

Sistema de Gestión de Pedido: Es el sistema que maneja el dueño del negocio para gestionar la atención al cliente.

***Pantalla Inicial/Login:*** Al iniciar el sistema los usuarios se encuentran con la pantalla de bienvenida del sistema. En la parte inferior de la pantalla se encuentran los campos Usuario y Contraseña, los cuales permitirán tener acceso al sistema solo los usuarios autorizados.



Figura x. : Login de la aplicación.

Al ingresar el usuario y su contraseña, se direccionara a la pantalla de pedidos, En caso contrario aparecerá un mensaje de "Usuario o contraseña incorrecto".

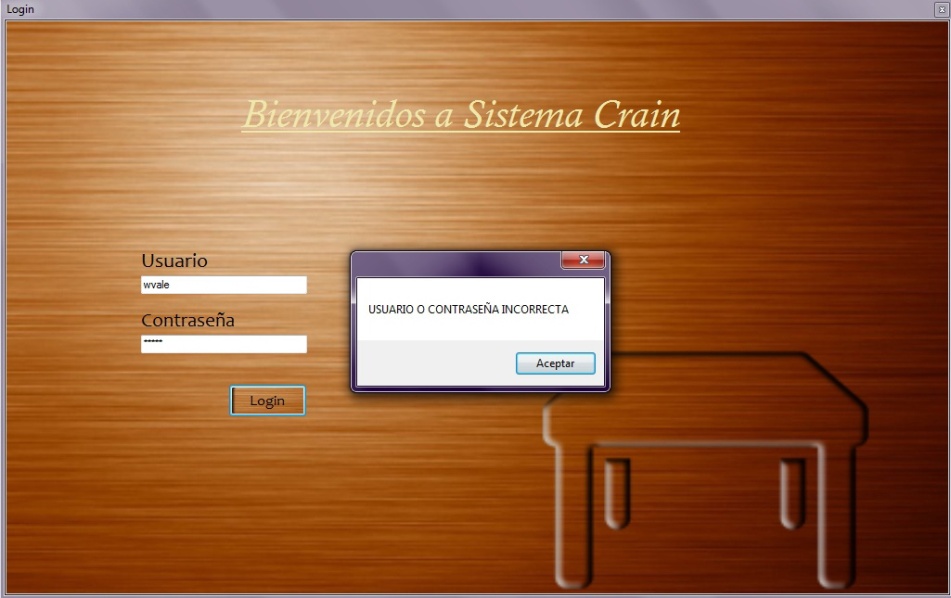


Figura x. : Autenticación incorrecta.

***Pantalla Pedidos:*** En esta pantalla se puede observar los pedidos que han sido enviados por los clientes. Cuenta con un cuadro principal que lista todos los pedidos con los siguientes datos: idPedido, Fecha, Total, Mesa.

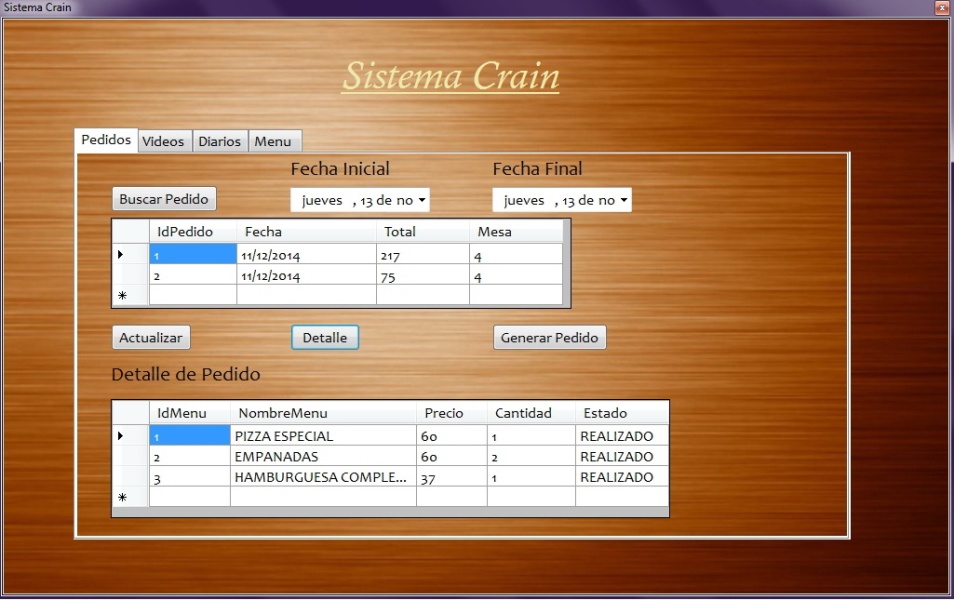


Figura x. : Pantalla principal de pedidos.

* Botón Buscar pedido: genera una lista con los pedidos desde una fecha inicial hasta una final y los muestra en el cuadro principal.
* Botón Actualizar: Actualiza la lista de pedidos de la pantalla principal.
* Botón Detalle: Muestra en detalle un pedido. Muestra la siguiente informacion: IdMenu, Nombre, Precio, Cantidad y Estado.
* Botón Generar Pedido: Permite al usuario del sistema imprimir un pedido en el directorio "CharlottePedidos".



Figura x. : Pedido generado por el sistema.

Pantalla Videos: En esta pantalla se observan los videos definidos por el administrador. El cuadro principal muestra la lista de direcciones de videos y se remarca con color el video al cual el usuario tendrá acceso. Cuenta con un cuadro principal que lista los videos con los siguientes datos: id, Nombre, Url y Estado.

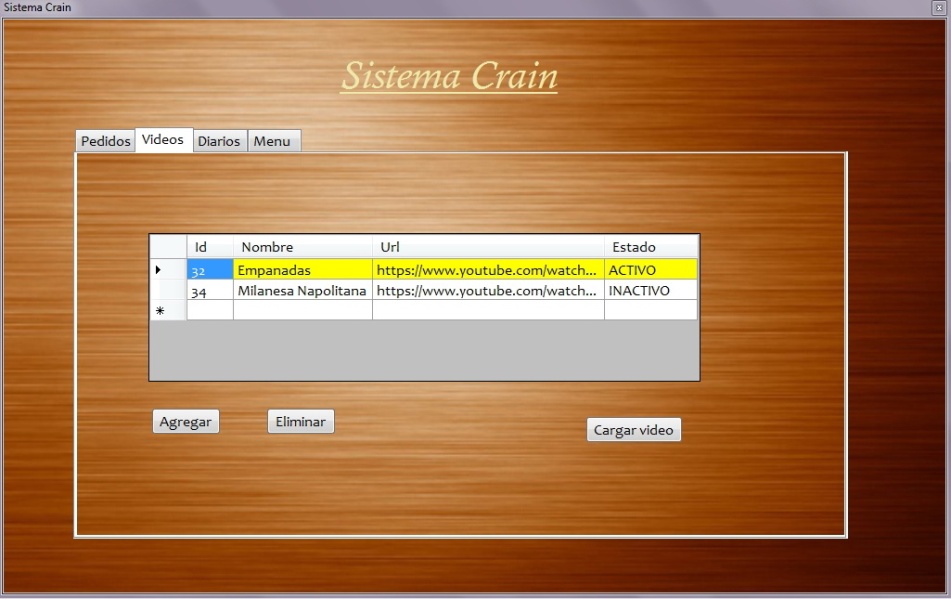


Figura x. : Pantalla principal de video.

* Botón Agregar: Agrega un nuevo video a la lista de video.

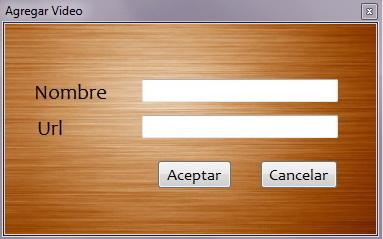


Figura x. : Pantalla Agregar video.

* Botón Eliminar: Elimina un video de la lista.
* Botón Cargar video: Selecciona el video que el cliente podrá ver desde la aplicación móvil.

***Pantalla Diarios:*** En esta pantalla se observan los diarios digitales definidos por el administrador. El cuadro principal muestra la lista de direcciones de los diarios digitales a los que el cliente tendrá acceso. Cuenta con un cuadro principal que lista los Diarios con los siguientes datos: id, Nombre y Url.

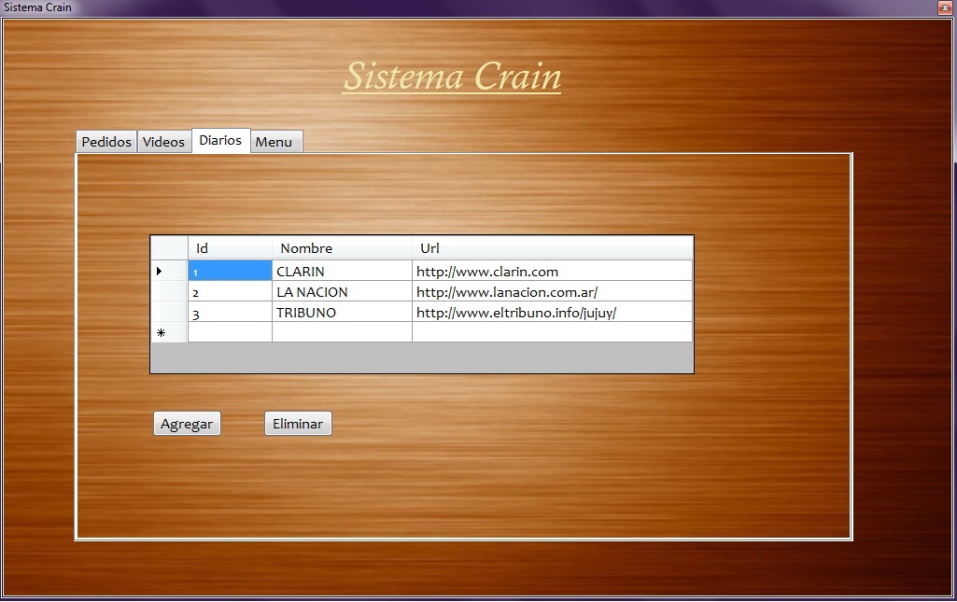


Figura x. : Pantalla principal de Diario.

* Botón Agregar: Agrega una nueva dirección de diario digital a la lista.



Figura x. : Pantalla Agregar Diario.

* Botón Eliminar: Elimina un dirección de diario digital de la lista.

***Pantalla Menú:*** En esta pantalla se observan los Menú definidos por el administrador. El cuadro principal muestra la lista de Menú que se mostraran en la Carta Gourmet. Cuenta con un cuadro principal que lista todos los Menú con los siguientes datos: id, Nombre, Precio, Descripción y Categoría .



Figura x. : Pantalla principal Menú.

* Botón Modificar Precio: Permite modificar el precio de uno de los Menú.



Figura x. : Pantalla Modificar Precio.

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 de "Sistema de Gestión de Pedido”.

Para el desarrollo de la aplicación de escritorio se utiliza la herramienta Visual estudio 2010. La codificación se realiza con lenguajes de c#.

## Estructura de paquetes: a continuación se presenta la distribución de paquetes en realizados en la codificación.

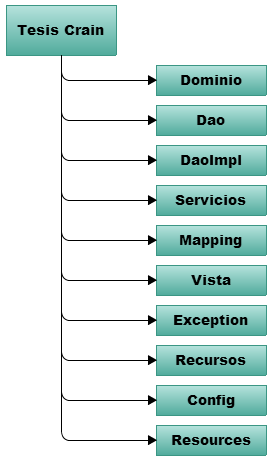


Figura x.x: Diagrama de paquetes de sistema Crain

A continuación se definen el contenido de cada uno de los paquetes:

**Dominio**: clases de entidades identificadas en el sistema. Contiene las clases:

* Administrador.cs
* Categoria.cs
* Diario.cs
* Menu.cs
* Mesa.cs
* Pedido.cs
* Reporte.cs
* TeaMenuPedido.cs
* Video.cs

**Dao**: son interfaces las cuales implementaran las clases. Las interfaces son:

* AdministradorDao.cs
* CategoriaDao.cs
* DiarioDao.cs
* MenuDao.cs
* MesaDao.cs
* PedidoDao.cs
* ReporteDao.cs
* TeaMenuPedidoDao.cs
* VideoDao.cs

**DaoImpl**: contiene las clases que implementan las interfaces definidas en el paquete dao. Las clases son:

* AdministradorDaoImpl.cs
* CategoriaDaoImpl.cs
* DiarioDaoImpl.cs
* MenuDaoImpl.cs
* MesaDaoImpl.cs
* PedidoDaoImpl.cs
* ReporteDaoImpl.cs
* TeaMenuPedidoDaoImpl.cs
* VideoDaoImpl.cs

**Exception**: clases que manejarán los errores.

* ValorNumericoException.cs

**Servicios**: se implementa el patrón facade

* Facade.cs

**Mapping**: se encuentran los archivos de mapeo de hibernate.

* administrador.hbm.xml
* categoria.hbm.xml
* diario.hbm.xml
* menu.hbm.xml
* mesa.hbm.xml
* pedido.hbm.xml
* reporte.hbm.xml
* teamenupedido.hbm.xml
* video.hbm.xml

**Vista**: contiene las pantallas del sistema

* Login.cs
* PantallaPrincipal.cs
* AgregarDiario.cs
* AgregarVideo.cs
* ModificarPrecio.cs

**Recursos:** archivos adicionales necesarios en visual estudio para lograr el mapeo.

* nhibernate-configuration.xsd
* nhibernate-mapping.xdsd

**Config:** contiene los archivos que permiten el **mapeo** entre entidades de la base de datos y los objetos de la aplicación.

* hibernate.cfg.xml
* NHibernateManager

**Resources:**  contiene las imágenes utilizadas en el sistema.

* numerMesa2-2.png
* textField2b.png

Elementos del dominio:

*Clase Menu:*

**public** class Menu

**{**

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

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

**private** double precioMenu**;**

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

**private** ISet teaMenuPedidoMap **=** **new** HashedSet**();**

**private** ISet complementoMap **=** **new** HashedSet**();**

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

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

**public** Menu**(**int idMenu**,** string nombrePlatoMenu**,** double precioMenu**,** string descripcionMenu**,** Categoria categoria**)** **{**

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

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

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

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

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

**}**

**}**

*Clase Pedido:*

**public** class Pedido

**{**

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

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

**private** double totalPedido**;**

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

**private** ISet teaMenuPedidoMap **=** **new** HashedSet**();**

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

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

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

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

**this.**totalPedido **=** totalPedido**;**

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

**}**

**}**

Elementos de Dao:

*Interfaz MenuDao:*

public interface MenuDao

{

void guardarMenu(Menu menu);

IList<Menu> getAllMenu();

void modificarMenu(Menu menu);

}

*Interfaz PedidoDao:*

public interface PedidoDao

{

void guardarPedido(Pedido pedido);

IList<Pedido> getAllPedido();

}

Elementos de DaoImpl:

*Clase MenuDaoImpl:*

public class MenuDaoImpl:MenuDao

{

public void guardarMenu(Menu menu) {

NHibernateManager manager = new NHibernateManager();

{

using (ITransaction transaction = manager.Session.BeginTransaction())

{

manager.Session.Save(menu);

transaction.Commit();

}

}

}

public void modificarMenu(Menu menu) {

NHibernateManager manager = new NHibernateManager();

{

using (ITransaction transaction = manager.Session.BeginTransaction())

{

manager.Session.Update(menu);

transaction.Commit();

}

}

}

public IList<Menu> getAllMenu() {

IList<Menu> lista = null;

NHibernateManager manager = new NHibernateManager();

using (ISession session = manager.Session)

{

lista = session.CreateCriteria(typeof(Menu)).List<Menu>();

session.Close();

}

return lista;

}

}

*Clase PedidoDaoImpl:*

public class PedidoDaoImpl : PedidoDao

{

public void guardarPedido(Pedido pedido)

{

NHibernateManager manager = new NHibernateManager();

{

using (ITransaction transaction = manager.Session.BeginTransaction())

{

manager.Session.Save(pedido);

transaction.Commit();

}

}

}

public IList<Pedido> getAllPedido()

{

IList<Pedido> lista = null;

NHibernateManager manager = new NHibernateManager();

using (ISession session = manager.Session)

{

lista = session.CreateCriteria(typeof(Pedido)).List<Pedido>();

session.Close();

}

return lista;

}

}

Elementos de Mapping:

*Mapeo de Menu:*

<class name="Menu" table="menu" lazy="true" >

<id name="IdMenu" column="MENU\_ID" type="int" unsaved- value="0">

<generator class="native"/>

</id>

<property name="NombrePlatoMenu" column="MENU\_NOMBRE\_PLATO" type="String">

</property>

<property name="PrecioMenu" column="MENU\_PRECIO" type="double">

</property>

<property name="DescripcionMenu" column="MENU\_DESCRIPCION" type="String">

</property>

<many-to-one name="Categoria" class="Categoria" fetch="join" lazy="false">

<column name="CATEGORIA\_ID" />

</many-to-one>

<set name="ComplementoMap" table="COMPLEMENTO" inverse="true" lazy="false" fetch="select">

<key>

<column name="MENU\_ID" />

</key>

<one-to-many class="ComplementoMenu" />

</set>

<set name="TeaMenuPedidoMap" table="tea\_menu\_pedido" inverse="true" lazy="false" fetch="select">

<key>

<column name="Menu\_ID" />

</key>

<one-to-many class="TeaMenuPedido" />

</set>

</class>

</hibernate-mapping>

*Mapeo de Pedido:*

<class name="Pedido" table="pedido" lazy="true" >

<id name="IdPedido" column="PEDIDO\_ID" type="int" unsaved- value="0">

<generator class="native"/>

</id>

<property name="FechaPedido" column="PEDIDO\_FECHA" type="String">

</property>

<property name="TotalPedido" column="PEDIDO\_TOTAL" type="double">

</property>

<many-to-one name="Mesa" class="Mesa" fetch="join" lazy="false">

<column name="MESA\_ID" />

</many-to-one>

<set name="TeaMenuPedidoMap" table="tea\_menu\_pedido" inverse="true" lazy="false" fetch="select">

<key>

<column name="PEDIDO\_ID" />

</key>

<one-to-many class="TeaMenuPedido" />

</set>

</class>

</hibernate-mapping>

Elemento de Servicio:

Facade

class Facade

{

...

MenuDaoImpl menuDaoImpl = new MenuDaoImpl();

PedidoDaoImpl pedidoDaoImpl = new PedidoDaoImpl();

...

public Facade() { }

...

//Servicios de Menu

public IList<Menu> getAllMenu()

{

return menuDaoImpl.getAllMenu();

}

public void modificarMenu(Menu menu)

{

this.menuDaoImpl.modificarMenu(menu);

}

// Servicio de Pedido

public IList<Pedido> getAllPedido()

{

return pedidoDaoImpl.getAllPedido();

}

...

}

}

Elementos de Vista:

Pantalla Login

public partial class Login : Form

{ Facade facade = new Facade();

private bool bandera = false;

public Login()

{

InitializeComponent();

}

private void buttonAceptar\_Click(object sender, EventArgs e)

{

if (facade.verificarAdministrador(textBoxUsuario.Text, textBoxPassword.Text))

{ bandera = true;

}

else

{ MessageBox.Show("USUARIO O CONTRASEÑA INCORRECTA");

}

if (bandera)

{

this.DialogResult = DialogResult.OK;

}

}

private void Login\_Load(object sender, EventArgs e)

{ }

private void label2\_Click(object sender, EventArgs e)

{ }

private void label1\_Click(object sender, EventArgs e)

{ }

private void label3\_Click(object sender, EventArgs e)

{ }

}

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#.

Estructura de directorio

A continuación se presenta la distribución de carpetas en el Entorno de Desarrollo Unity 3D.

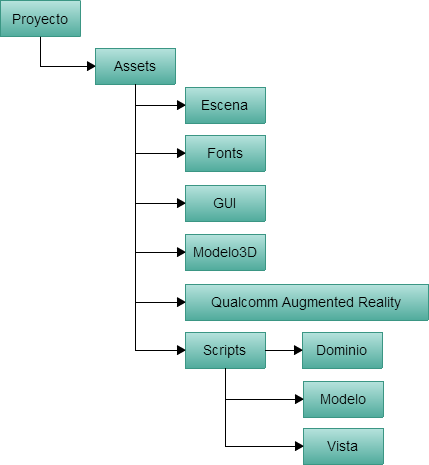


Figura 6.X Estructura de Directorio “Sistema de Atención Aumentada”.

Assets: Es la carpeta raíz de los Asstes de la aplicación. Dentro de esta carpeta se puede importar objetos 3D, texturas, fonts, prefabs, scripts, etc.

Escena: Contiene la Escena llamada “Sistema Atención Aumentada”. Este un archivo que guarda información de los objetos que actualmente están activos en la aplicación.

Fonts: Almacena los tipo de letra que usa en la aplicación. Están en formato ttf.

GUI: Contiene los botones, box, icono y splash de la aplicación. Estos archivos están en formato png.

Modelo3D: Almacena los objetos 3D de la aplicación. Estos objetos están en formato 3DS y obj.

Qualcomm Augmented Reality: Esta contiene las librerías de Realidad Aumentada necesarias para el desarrollo del prototipo. Dispone de Scripts, prefabs, Textures, Fonts, Shaders y Materials ya predefinidos y listo para usar.

Scripts: Contiene la totalidad de los scripts desarrollados.

Dominio: clases de entidades identificadas en el sistema. Esta carpeta contiene las siguientes clases:

* Menu
* Categoria
* Mesa
* Diario
* Pedido
* TeaMenuPedido

Modelo:

* GuardarPedidoBD: Esta clase permite dar de alta a un pedido
* AppIUConstantes: Define todas las constantes utilizas en la aplicación
* NumeroMesa: Permite obtener una lista con objetos de tipo Mesa. Son traídas de la base de datos alojada en el servidor.
* AgregarMenuPedido: Esta clase permite armar el detalle del pedido. Este detalle es almacenado en un ArrayList llamado listaTeaMenuPedido.
* ReproducirVideo: Esta clase permite tener acceso a los videos de recomendación del chef. Son traídas de la base de datos alojada en el servidor.
* HabilitarPantallaServidor: Esta clase permite cargar un lista con objetos de tipo Menu. Son traídas de la base de datos alojada en el servidor.
* DefaultTrackableEventHandler: Implementa las funciones Encontrado y Perdido de un marcador.

Vista: Esta carpeta contiene los scripts necesarios para dibujar la interfaz de la aplicación. Las clases que contiene son las siguientes:

* AdministracionServidor: Permite visualizar la interfaz de configuración del servidor.
* RedimencionarPantalla: Esta clase dibuja los botones Agregar Pedido, Ver Perdido y Navegación en la pantalla de la cámara del dispositivo. Reubica y redimensiona los botones en función del tamaño de la pantalla.
* DibujarPantalla: Esta clase permite visualizar la interfaz de Pedido.
* DiarioDigital: Crea la interfaz de Diario digital
* VisualizadorModelo3D: Esta clase permite visualizar los modelos 3D en la carta gourmet .
* About: Permite dibujar la interfaz del about.
* CambiarColor: Permite cambiar el color del texto de la letras aumentadas.

La aplicación móvil “Sistema de Atención Aumentada” trabaja con el lenguaje PHP para solicitar servicios al servidor local. El servidor local recibe la petición, reúne la información necesaria consultando las bases de datos y responde con la información solicitada. Los archivos .php generados para esta aplicación son:

* connect\_bd.php: Permite abrir la conexión de la base de datos Mysql.
* guardar\_pedido.php: Recibe como parámetro el encabezado de un pedido y lo almacena en la base de datos.
* guardar\_tea\_menu\_pedido.php: Recibe como parámetro el detalle de un pedido y lo almacena en la base de datos.
* indice\_mayor\_pedido.php: Obtiene el id de mayor valor de todos los pedidos almacenados.
* select\_diario.php: Obtiene los datos de la tabla Diario.
* select\_mesa.php: Obtiene los datos de la tabla Mesa.
* select\_video\_activo.php: Obtiene una fila de la tabla video, que tenga el atributo estado en “Activo”.
* selectMenu.php: Obtiene los datos de la tabla Menu

Para crear la base de datos, se uso phpMyAdmin, un gestor de base de datos Mysql a través de pagina web. Se exporto esta base con el siguiente nombre:

* dbms\_crain.sql: este contiene las estructura de las tablas y los datos bases asociados a cada una de tabla. Este archivo se utiliza para importar la base de datos a otro servidor en caso que se lo requiera. La base de datos se utiliza tanto para el “Sistema de Atención Aumentada” como para el “Sistema de Gestión de Pedidos”.

A continuación se muestra los código más relevante del Sistema de Atención Aumentada.

**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 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 DefaultTrackableEventHandler**

**using** UnityEngine**;**

//Implementa las funciones Encontrado y Perdido para la detección de //marcadores.

**public** class DefaultTrackableEventHandler **:** MonoBehaviour**,**

ITrackableEventHandler

**{**

#region PRIVATE\_MEMBER\_VARIABLES

**private** TrackableBehaviour mTrackableBehaviour**;**

**private** bool banderaHabilitarPantallaDeNavegacion**=false;**

//private EventosMarcador eventoMarcador;

#endregion // PRIVATE\_MEMBER\_VARIABLES

#region UNTIY\_MONOBEHAVIOUR\_METHODS

void Start **()**

**{**

mTrackableBehaviour **=** GetComponent**<**TrackableBehaviour**>** **();**

**if** **(**mTrackableBehaviour**)** **{**

mTrackableBehaviour**.**RegisterTrackableEventHandler **(this);**

**}**

**}**

#endregion // UNTIY\_MONOBEHAVIOUR\_METHODS

#region PUBLIC\_METHODS

/// <summary>

/// Implementation of the ITrackableEventHandler function called when the

/// tracking state changes.

/// </summary>

**public** void OnTrackableStateChanged **(**

TrackableBehaviour**.**Status previousStatus**,**

TrackableBehaviour**.**Status newStatus**)**

**{**

**if** **(**newStatus **==** TrackableBehaviour**.**Status**.**DETECTED **||**

newStatus **==** TrackableBehaviour**.**Status**.**TRACKED**)** **{**

OnTrackingFound **();**

**this.**banderaHabilitarPantallaDeNavegacion**=true;**

**}** **else** **{**

**this.**banderaHabilitarPantallaDeNavegacion**=false;**

OnTrackingLost **();**

**}**

**}**

#endregion // PUBLIC\_METHODS

#region PRIVATE\_METHODS

**private** void OnTrackingFound **()**

**{**

Renderer**[]** rendererComponents **=** GetComponentsInChildren**<**Renderer**>** **(true);**

Collider**[]** colliderComponents **=** GetComponentsInChildren**<**Collider**>** **(true);**

// Enable rendering:

**foreach** **(**Renderer component **in** rendererComponents**)** **{**

component**.**enabled **=** **true;**

**}**

// Enable colliders:

**foreach** **(**Collider component **in** colliderComponents**)** **{**

component**.**enabled **=** **true;**

**}**

Debug**.**Log **(**"Trackable " **+** mTrackableBehaviour**.**TrackableName **+** " found"**);**

**}**

**private** void OnTrackingLost **()**

**{**

Renderer**[]** rendererComponents **=** GetComponentsInChildren**<**Renderer**>** **(true);**

Collider**[]** colliderComponents **=** GetComponentsInChildren**<**Collider**>** **(true);**

// Disable rendering:

**foreach** **(**Renderer component **in** rendererComponents**)** **{**

component**.**enabled **=** **false;**

**}**

// Disable colliders:

**foreach** **(**Collider component **in** colliderComponents**)** **{**

component**.**enabled **=** **false;**

**}**

Debug**.**Log **(**"Trackable " **+** mTrackableBehaviour**.**TrackableName **+** " lost"**);**

**}**

**public** bool getBanderaHabilitarPantallaDeNavegacion**(){**

**return** **this.**banderaHabilitarPantallaDeNavegacion**;**

**}**

**public** void getBanderaHabilitarPantallaDeNavegacion**(**bool bandera**){**

**this.**banderaHabilitarPantallaDeNavegacion**=**bandera**;**

**}**

#endregion // PRIVATE\_METHODS

**}**

**Redimensionar Pantalla:**

**using** UnityEngine**;**

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

**using** System**;**

//Este clase permite redimensionar los botones según el tamaño de la //pantalla. Los botones son Ver Pedido, Agregar Pedido y Navegación.

**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**;**

**}**

**}**

**connect\_bd.php**

**//Permite abrir la conexión la base de datos Mysql**

<?php

$link=mysql\_connect("localhost", "root","crain"); // abro una conexion

**if**($link){

mysql\_select\_db("dbms\_crain",$link); // selecciono la base de datos.

}

?>

**guardar\_pedido.php**

**//Recibe como parámetro el encabezado de un pedido y lo almacena en la base de datos.**

<?php

$idPedido = $\_GET['idPedido'];

$fecha = $\_GET['fecha'];

$totalAPagar = $\_GET['totalAPagar'];

$idMensa = $\_GET['idMesa'];

**require**("connect\_bd.php");

$q="INSERT INTO Pedido (PEDIDO\_ID,PEDIDO\_FECHA,PEDIDO\_TOTAL,MESA\_ID) VALUES (".$idPedido.",'".$fecha."',".$totalAPagar.",".$idMensa.") ;";

$result= mysql\_query($q, $link) **or** **die** ("problema con query");

?>

**guardar\_tea\_menu\_pedido.php**

**//Recibe como parámetro el detalle de un pedido y lo almacena en la base de datos.**

<?php

$idTea = $\_GET['idTea'];

$cantidadTea = $\_GET['cantidadTea'];

$idMenu = $\_GET['idMenu'];

$estadoTea = $\_GET['estadoTea'];

$idPedido = $\_GET['idPedido'];

**require**("connect\_bd.php");

$q="INSERT INTO tea\_menu\_pedido (TEA\_ID, TEA\_CANTIDAD, TEA\_ESTADO,PEDIDO\_ID, MENU\_ID) VALUES (".$idTea.",".$cantidadTea.",'".$estadoTea."',".$idPedido.",".$idMenu.") ;";

$result= mysql\_query($q, $link) **or** **die** ("problema con query");

mysql\_close($link);

?>

**indice\_mayor\_pedido.php**

**// Obtiene el id de mayor valor de todos los pedidos almacenados.**

<?php

**require**("connect\_bd.php");

$query = "SELECT MAX(PEDIDO\_ID) AS PEDIDO\_ID FROM pedido";

$result = mysql\_query($query,$link);

$row = mysql\_fetch\_row($result) **or** **die**(mysql\_error());

**echo** $row[0];

?>

**select\_diario.php**

**//Obtiene los datos de la tabla Diario.**

<?php

**require**("connect\_bd.php");

$result = mysql\_query("SELECT \* FROM diario");

$indice=-1;

**while** ($fila = mysql\_fetch\_assoc($result)) {

$indice=$indice+1;

$vector[$indice]=$fila['DIARIO\_ID'];

$indice=$indice+1;

$vector[$indice]=$fila['DIARIO\_NOMBRE'];

$indice=$indice+1;

$vector[$indice]=$fila['DIARIO\_URL'];

}

**die**(implode("->",$vector));

mysql\_close($link);

?>

**select\_mesa.php**

**//Obtiene los datos de la tabla Mesa.**

<?php

**require**("connect\_bd.php");

$result = mysql\_query("SELECT \* FROM mesa");

$indice=-1;

**while** ($fila = mysql\_fetch\_assoc($result)) {

$indice=$indice+1;

$vector[$indice]=$fila['MESA\_ID'];

$indice=$indice+1;

$vector[$indice]=$fila['MESA\_DESCRIPCION'];

}

**die**(implode("->",$vector));

mysql\_close($link);

?>

**select\_video\_activo.php**

//Obtiene una fila de la tabla video, que tenga el atributo estado en “Activo”.

<?php

**require**("connect\_bd.php");

$query = "SELECT VIDEO\_URL FROM video WHERE VIDEO\_ESTADO = 'ACTIVO'";

$result = mysql\_query($query,$link);

$row = mysql\_fetch\_row($result) **or** **die**(mysql\_error());

**echo** $row[0];

mysql\_close($link);

?>

**selectMenu.php**

//Obtiene los datos de la tabla Menú

<?php

**require**("connect\_bd.php");

$result = mysql\_query("SELECT \* FROM menu");

$indice=-1;

**while** ($fila = mysql\_fetch\_assoc($result)) {

$indice=$indice+1;

$vector[$indice]=$fila['MENU\_ID'];

$indice=$indice+1;

$vector[$indice]=$fila['MENU\_NOMBRE\_PLATO'];

$indice=$indice+1;

$vector[$indice]=$fila['MENU\_PRECIO'];

$indice=$indice+1;

$vector[$indice]=$fila['MENU\_DESCRIPCION'];

$indice=$indice+1;

$vector[$indice]=$fila['CATEGORIA\_ID'];

}

**die**(implode("->",$vector));

mysql\_close($link);

?>

**dbms\_crain.sql**

--La base de datos se utiliza tanto para el Sistema de Atención Aumentada

--como para el Sistema de Gestión de Pedidos.

**SET** SQL\_MODE **=** "NO\_AUTO\_VALUE\_ON\_ZERO"**;**

**SET** time\_zone **=** "+00:00"**;**

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/**;**

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/**;**

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/**;**

/\*!40101 SET NAMES utf8 \*/**;**

--

-- Base de datos: `dbms\_crain`

--

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `administrador`

--

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `administrador` **(**

`ADMINISTRADOR\_ID` **int(**10**)** **NOT** **NULL** AUTO\_INCREMENT**,**

`ADMINISTRADOR\_NOMBRE` **varchar(**100**)** **NOT** **NULL,**

`ADMINISTRADOR\_APELLIDO` **varchar(**100**)** **NOT** **NULL,**

`ADMINISTRADOR\_USUARIO` **varchar(**100**)** **NOT** **NULL,**

`ADMINISTRADOR\_PASSWORD` **varchar(**100**)** **NOT** **NULL,**

`ADMINISTRADOR\_CARGO` **varchar(**100**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`ADMINISTRADOR\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**3 **;**

--

-- Volcado de datos para la tabla `administrador`

--

**INSERT** **INTO** `administrador` **(**`ADMINISTRADOR\_ID`**,** `ADMINISTRADOR\_NOMBRE`**,** `ADMINISTRADOR\_APELLIDO`**,** `ADMINISTRADOR\_USUARIO`**,** `ADMINISTRADOR\_PASSWORD`**,** `ADMINISTRADOR\_CARGO`**)** **VALUES**

**(**1**,** 'Martin'**,** 'Carrizo'**,** 'mcarrizo'**,** 'carrizocrain'**,** 'empleado'**),**

**(**2**,** 'Walter'**,** 'Vale'**,** 'wvale'**,** 'valecrain'**,** 'empleado'**);**

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `categoria`

--

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `categoria` **(**

`CATEGORIA\_ID` **int(**10**)** **NOT** **NULL** AUTO\_INCREMENT**,**

`CATEGORIA\_NOMBRE` **varchar(**100**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`CATEGORIA\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**3 **;**

--

-- Volcado de datos para la tabla `categoria`

--

**INSERT** **INTO** `categoria` **(**`CATEGORIA\_ID`**,** `CATEGORIA\_NOMBRE`**)** **VALUES**

**(**1**,** 'PLATOS'**),**

**(**2**,** 'BEBIDAS'**);**

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `complemento`

--

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `complemento` **(**

`COMPLEMENTO\_ID` **int(**10**)** **NOT** **NULL** AUTO\_INCREMENT**,**

`COMPLEMENTO\_NOMBRE` **varchar(**100**)** **NOT** **NULL,**

`COMPLEMENTO\_DESCRIPCION` **varchar(**100**)** **NOT** **NULL,**

`MENU\_ID` **int(**10**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`COMPLEMENTO\_ID`**),**

**KEY** `MENU\_ID` **(**`MENU\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**1 **;**

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `diario`

--

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `diario` **(**

`DIARIO\_ID` **int(**10**)** **NOT** **NULL** AUTO\_INCREMENT**,**

`DIARIO\_NOMBRE` **varchar(**100**)** **NOT** **NULL,**

`DIARIO\_URL` **varchar(**600**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`DIARIO\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**7 **;**

--

-- Volcado de datos para la tabla `diario`

--

**INSERT** **INTO** `diario` **(**`DIARIO\_ID`**,** `DIARIO\_NOMBRE`**,** `DIARIO\_URL`**)** **VALUES**

**(**1**,** 'CLARIN'**,** 'http://www.clarin.com'**),**

**(**2**,** 'LA NACION'**,** 'http://www.lanacion.com.ar/'**),**

**(**3**,** 'TRIBUNO'**,** 'http://www.eltribuno.info/jujuy/'**);**

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `menu`

--

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `menu` **(**

`MENU\_ID` **int(**10**)** **NOT** **NULL** AUTO\_INCREMENT**,**

`MENU\_NOMBRE\_PLATO` **varchar(**100**)** **NOT** **NULL,**

`MENU\_PRECIO` **float(**10**,**2**)** **NOT** **NULL,**

`MENU\_DESCRIPCION` **varchar(**100**)** **NOT** **NULL,**

`CATEGORIA\_ID` **int(**10**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`MENU\_ID`**),**

**KEY** `CATEGORIA\_ID` **(**`CATEGORIA\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**11 **;**

--

-- Volcado de datos para la tabla `menu`

--

**INSERT** **INTO** `menu` **(**`MENU\_ID`**,** `MENU\_NOMBRE\_PLATO`**,** `MENU\_PRECIO`**,** `MENU\_DESCRIPCION`**,** `CATEGORIA\_ID`**)** **VALUES**

**(**1**,** 'PIZZA ESPECIAL'**,** 60.00**,** 'DESCRIPCION'**,** 1**),**

**(**2**,** 'EMPANADAS'**,** 60.00**,** 'DESCRIPCION'**,** 1**),**

**(**3**,** 'PICADA'**,** 40.00**,** 'DESCRIPCION'**,** 1**),**

**(**4**,** 'HAMBURGUESA COMPLETA'**,** 37.00**,** 'DESCRIPCION'**,** 1**),**

**(**5**,** 'DESAYUNO AMERICANO'**,** 30.00**,** 'DESCRIPCION'**,** 1**),**

**(**6**,** 'LICUADO DE FRUTILLA'**,** 25.00**,** 'DESCRIPCION'**,** 2**),**

**(**7**,** 'JUGO DE NARANJA'**,** 25.00**,** 'DESCRIPCION'**,** 2**),**

**(**8**,** 'CAPUCHINO'**,** 25.00**,** 'DESCRIPCION'**,** 2**),**

**(**9**,** 'COCA COLA DE 350 CC'**,** 20.00**,** 'DESCRIPCION'**,** 2**),**

**(**10**,** 'VINO TINTO'**,** 30.00**,** 'DESCRIPCION'**,** 2**);**

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `mesa`

--

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `mesa` **(**

`MESA\_ID` **int(**10**)** **NOT** **NULL** AUTO\_INCREMENT**,**

`MESA\_DESCRIPCION` **varchar(**100**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`MESA\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**5 **;**

--

-- Volcado de datos para la tabla `mesa`

--

**INSERT** **INTO** `mesa` **(**`MESA\_ID`**,** `MESA\_DESCRIPCION`**)** **VALUES**

**(**1**,** 'ADELANTE A LA DERECHA'**),**

**(**2**,** 'ADELANTE A LA IZQUEIRDA'**),**

**(**3**,** 'ATRAS A LA IZQUIERDA'**),**

**(**4**,** 'ATRAS A LA DERECHA'**);**

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `pedido`

--

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `pedido` **(**

`PEDIDO\_ID` **int(**10**)** **NOT** **NULL** AUTO\_INCREMENT**,**

`PEDIDO\_FECHA` **varchar(**100**)** **NOT** **NULL,**

`PEDIDO\_TOTAL` **float(**10**,**2**)** **NOT** **NULL,**

`MESA\_ID` **int(**10**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`PEDIDO\_ID`**),**

**KEY** `MESA\_ID` **(**`MESA\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**109 **;**

--

-- Volcado de datos para la tabla `pedido`

--

**INSERT** **INTO** `pedido` **(**`PEDIDO\_ID`**,** `PEDIDO\_FECHA`**,** `PEDIDO\_TOTAL`**,** `MESA\_ID`**)** **VALUES**

**(**1**,** '11/12/2014'**,** 217.00**,** 4**),**

**(**2**,** '11/12/2014'**,** 75.00**,** 4**);**

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `tea\_menu\_pedido`

--

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `tea\_menu\_pedido` **(**

`TEA\_ID` **int(**10**)** **NOT** **NULL** AUTO\_INCREMENT**,**

`TEA\_CANTIDAD` **int(**10**)** **NOT** **NULL,**

`TEA\_ESTADO` **varchar(**100**)** **NOT** **NULL,**

`PEDIDO\_ID` **int(**10**)** **NOT** **NULL,**

`MENU\_ID` **int(**10**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`TEA\_ID`**),**

**KEY** `PEDIDO\_ID` **(**`PEDIDO\_ID`**),**

**KEY** `MENU\_ID` **(**`MENU\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**149 **;**

--

-- Volcado de datos para la tabla `tea\_menu\_pedido`

--

**INSERT** **INTO** `tea\_menu\_pedido` **(**`TEA\_ID`**,** `TEA\_CANTIDAD`**,** `TEA\_ESTADO`**,** `PEDIDO\_ID`**,** `MENU\_ID`**)** **VALUES**

**(**1**,** 1**,** 'REALIZADO'**,** 1**,** 1**),**

**(**2**,** 2**,** 'REALIZADO'**,** 1**,** 2**),**

**(**3**,** 1**,** 'REALIZADO'**,** 1**,** 4**),**

**(**4**,** 1**,** 'REALIZADO'**,** 2**,** 6**),**

**(**5**,** 1**,** 'REALIZADO'**,** 2**,** 7**),**

**(**6**,** 1**,** 'REALIZADO'**,** 2**,** 8**);**

-- --------------------------------------------------------

--

-- Estructura de tabla para la tabla `video`

--

**CREATE** **TABLE** **IF** **NOT** **EXISTS** `video` **(**

`VIDEO\_ID` **int(**10**)** **NOT** **NULL** AUTO\_INCREMENT**,**

`VIDEO\_NOMBRE` **varchar(**100**)** **NOT** **NULL,**

`VIDEO\_URL` **varchar(**600**)** **NOT** **NULL,**

`VIDEO\_ESTADO` **varchar(**50**)** **NOT** **NULL,**

**PRIMARY** **KEY** **(**`VIDEO\_ID`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**35 **;**

--

-- Volcado de datos para la tabla `video`

--

**INSERT** **INTO** `video` **(**`VIDEO\_ID`**,** `VIDEO\_NOMBRE`**,** `VIDEO\_URL`**,** `VIDEO\_ESTADO`**)** **VALUES**

**(**32**,** 'Empanadas'**,** 'https://www.youtube.com/watch?v=-fRpz3bis4I'**,** 'ACTIVO'**),**

**(**34**,** 'Milanesa Napolitana'**,** 'https://www.youtube.com/watch?v=vjGLlxGKYjc'**,** 'INACTIVO'**);**

--

-- Restricciones para tablas volcadas

--

--

-- Filtros para la tabla `complemento`

--

**ALTER** **TABLE** `complemento`

**ADD** **CONSTRAINT** `complemento\_ibfk\_1` **FOREIGN** **KEY** **(**`MENU\_ID`**)** **REFERENCES** `menu` **(**`MENU\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE;**

--

-- Filtros para la tabla `menu`

--

**ALTER** **TABLE** `menu`

**ADD** **CONSTRAINT** `menu\_ibfk\_1` **FOREIGN** **KEY** **(**`CATEGORIA\_ID`**)** **REFERENCES** `categoria` **(**`CATEGORIA\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE;**

--

-- Filtros para la tabla `pedido`

--

**ALTER** **TABLE** `pedido`

**ADD** **CONSTRAINT** `pedido\_ibfk\_1` **FOREIGN** **KEY** **(**`MESA\_ID`**)** **REFERENCES** `mesa` **(**`MESA\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE;**

--

-- Filtros para la tabla `tea\_menu\_pedido`

--

**ALTER** **TABLE** `tea\_menu\_pedido`

**ADD** **CONSTRAINT** `tea\_menu\_pedido\_ibfk\_1` **FOREIGN** **KEY** **(**`PEDIDO\_ID`**)** **REFERENCES** `pedido` **(**`PEDIDO\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE,**

**ADD** **CONSTRAINT** `tea\_menu\_pedido\_ibfk\_2` **FOREIGN** **KEY** **(**`MENU\_ID`**)** **REFERENCES** `menu` **(**`MENU\_ID`**)** **ON** **DELETE** **CASCADE** **ON** **UPDATE** **CASCADE;**

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/**;**

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/**;**

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/**;**

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. |