

Universidad Nacional de Colombia Sede Medellín

Facultad de Minas

Documentación Aplicación CASE (Computer Aided Software Engineering)

“MenúApp”

Presentado a:

Prof. Fernán Alonso Villa Garzón

Teoría de La Computación y la Compilación

Semestre 2020 – II

Por:

Juan Fernando Patiño Castro

Juan David Ordoñez Noriega

Juan David Mejía Ardila

David Santiago Espíndola Carvajal

Sergio Andrés Zambrano González

Esteban Rojas Henao

Medellin, 18 de Noviembre de 2020

Tabla de Contenido

Documentación Menú App

1. Introducción

Dentro del desarrollo de la asignatura “Teoría de la Computación y la Compilación” fue designado el desarrollo de una aplicación CASE (Computer Aided Software Engineering) aplicando todos los conceptos aprendidos en transcurso del semestre. La aplicación debía estar en capacidad de traducir de forma automática un esquema preconceptual a una aplicación funcional.

Para lograrlo, este equipo propone ingresar dicho esquema en formato xml, dentro de la aplicación es procesado y, finalmente convertido en una aplicación web desarrollada en Javascript. Para ejemplificar su uso se escoge el menú de un restaurante, donde se añaden los platos y las bebidas, adicionalmente es posible añadir la información relacionada al restaurante. La aplicación utiliza una base de datos para el almacenamiento de la información de cada concepto clase creado y para gestionar el acceso a dicha información que cada tipo de usuario o rol (administrador, cliente, invitado, etc) puede tener.

2. Reglas de Conversión

2.1. Reglas Para la Creación del Esquema Preconceptual

1. El elemento “Rectangle” de la sección “General” en draw.io se traduce como una entidad o un atributo de la misma entidad. Las entidades se identifican por que de ellas se origina una única conexión hacia una relación estructural, de la cual a su vez salen múltiples conexiones a otros rectángulos correspondientes a los atributos de dicha entidad.

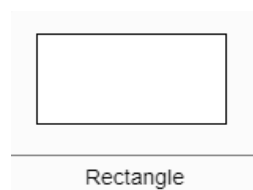


Figura 1. Elemento “Rectangle”.

2. El elemento “Start/ End” de la sección “Data flow Diagram” en Draw.io se traduce como una relación estructural. A pesar de **no ser necesario**, lo ideal es que en su interior se escriba la palabra “Tiene”. Este elemento conecta las entidades con sus respectivos atributos.



Figura 2. Elemento “Start/End”.

3. El elemento “Actor” de la sección “General” en draw.io se traduce como un usuario y designa su rol. El rol corresponde al nombre asignado al dar doble click sobre él y este aparece en la página principal de la aplicación final, en donde se escoge el rol deseado para interactuar en la aplicación.

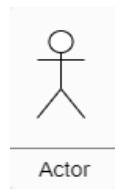


Figura 3. Elemento “Actor”.

4. El elemento “Derived Attribute” de la sección “Entity Relation” en draw.io se traduce como una relación dinámica. Este elemento **admite únicamente las siguientes palabras específicas**: “Actualizar”, “Ver”, “Borrar” o “Insertar”, las cuales corresponden a las acciones que puede realizar un actor sobre alguna entidad. No se habilitará ninguna opción si se usa un verbo distinto a estos cuatro, se debe tener en cuenta las letras mayúsculas.

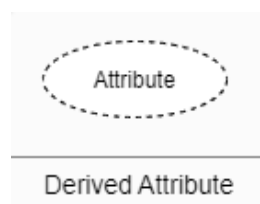


Figura 4. Elemento “Derived Attribute”.

5. Todo elemento del esquema debe estar enlazado por medio de flechas. Estas uniones deben realizarse en un sentido específico como se indica a continuación:
 - a. Empieza en una entidad y termina en una relación estructural.

- b. Empieza en una relación estructural y termina en un atributo.
 - c. Empieza en el actor y finaliza en una relación dinámica.
 - d. Empieza en una relación dinámica y finaliza en una entidad.
6. En el esquema preconceptual no se van a inicializar variables, solo van a ser declaradas.
 7. Debe existir una relación dinámica (“Actualizar”, “Ver”, “Borrar” o “Insertar”) por cada actor que deba tener acceso. Es decir, no se pueden compartir entre actores, p. ej: no se puede usar el mismo ovalo con la palabra “Actualizar” para un usuario Invitado y administrador.

2.2. Reglas Para Pasar del preconceptual al formato xml

1. Una vez creado el esquema, se debe exportar **únicamente** la página en donde fue creado, el archivo se descarga en formato .xml (es el único que se acepta como entrada en la aplicación). Notar que las opciones “compressed”, “selection only” y “all pages” que ofrece draw.io deben encontrarse deshabilitadas. Esto se ejemplifica en la siguiente figura.

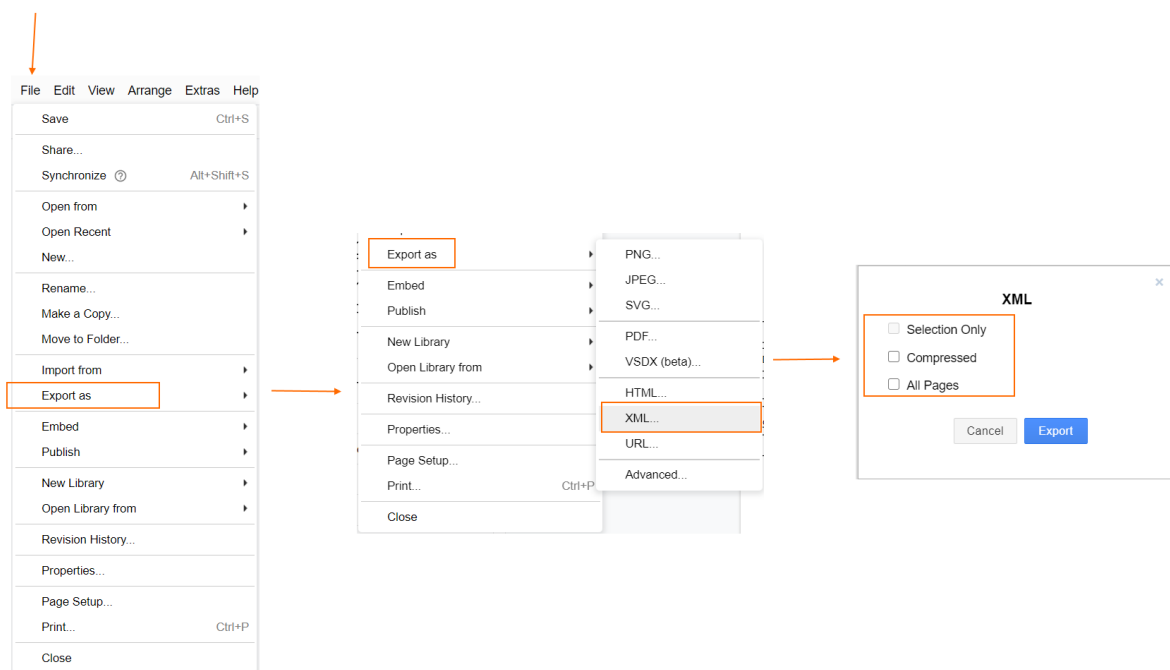


Figura 5. Ejemplo de cómo descargar el esquema creado en formato xml.

2. Si se agrega un elemento que no haya sido declarado en el esquema original, siguiendo las primeras 5 reglas, al ingresar el archivo a la aplicación aparecerá el siguiente mensaje de advertencia: “Se detectó un elemento desconocido: [Nombre Elemento]”.
3. El archivo xml se transforma a un formato json usando un parseo y se utilizan diversos algoritmos para vincular cada entidad con sus relaciones estructurales (atributos) y cada usuario con sus relaciones dinámicas (acciones) correspondientes.

2.3. Reglas Para Pasar del Json a la Aplicación

1. Se genera una interfaz con los usuarios creados y un link para compartir la aplicación. Adicionalmente se crea otra interfaz, para cada usuario, en donde se muestra cada concepto y las relaciones dinámicas asociadas representadas en forma de botones. Esto se obtiene en forma de aplicativo web en HTML, CSS y JavaScript.
2. Para todas las relaciones dinámicas que no hayan sido especificadas o asignadas entre un usuario y una entidad, causan que en su interfaz los botones asociados a cada entidad aparezcan de un color más pálido (azul claro), lo que significa, que está deshabilitada y no puede ser usada.
3. Se utiliza una base de datos no relacional usando MongoDB.
4. Se utiliza NodeJS como servidor para ejecutar la aplicación.

2.4. Reglas Para Uso de la Aplicación

1. Una vez se suba un nuevo esquema, la base de datos es borrada para poder recibir los datos del nuevo esquema, es decir, se guardan los datos de un único esquema (el último que haya sido utilizado).
2. Usando el link generado en la primera interfaz se podrá acceder de nuevo al último aplicativo creado usando la base de datos.
3. El botón “Volver” ubicado en la interfaz de alguno de los usuarios regresa al menú de usuarios (menú principal), donde se puede cambiar de usuario.
4. El botón “Insertar” conduce a una nueva página en donde es posible añadir los atributos de la entidad asociada. Aquí aparecen dos nuevos botones: “Guardar Información” y

“Terminar”, que permiten actualizar los cambios realizados y volver al menú del usuario, respectivamente. Es posible hacer la creación de una instancia (documento, en la base de datos no relacional) sin necesidad de haber añadido ningún atributo, ya que ningún campo es obligatorio.

5. El botón “Actualizar” conduce a una nueva página en donde se puede seleccionar una instancia de alguna entidad creada anteriormente. Se debe dar click sobre su tarjeta (cambia de color a naranja) y posteriormente sobre el botón “Seleccionar”, el cual nos lleva a nueva página en donde todos los campos aparecen vacíos, siendo necesario modificar únicamente los atributos que se deseen, de la instancia de la entidad. Aquellos no modificados mantienen el valor asignado anteriormente. En la página de “Actualizar”, existe también el botón “Volver” para regresar al menú principal.
6. El botón “Borrar” conduce a una nueva página en donde se puede seleccionar una instancia de la entidad creada anteriormente. Se debe dar click sobre su tarjeta (cambia de color a naranja) y posteriormente sobre el botón “Borrar”, eliminándola tanto de la base de datos como del aplicativo. En la página de “Borrar”, existe también el botón “Volver” para regresar al menú principal.
7. El botón “Ver” conduce a una nueva página en donde se pueden observar todas las instancias creadas de la entidad correspondiente. Aquí también aparece un botón “Volver” para regresar al menú del usuario.

3. Reglas de Definición de Requisitos Funcionales y No Funcionales

3.1. Requisitos Funcionales

1. La aplicación contará con una página web.
2. Permite cargar un archivo xml, el cual contiene los datos relacionados al esquema preconceptual.
3. Se generan los botones de las acciones correspondientes para cada tipo de usuario.

4. El aplicativo genera un formulario y permite insertar un documento de una entidad con sus respectivos atributos. Por ejemplo, insertar un usuario con su nombre, cédula, etc.
5. El aplicativo genera un “listado” de la(s) colección(nes) y permite actualizar la información de un documento seleccionado.
6. El aplicativo genera un “listado” de la(s) colección(nes) y permite borrar la información de un documento seleccionado.
7. El aplicativo permite consultar y ver los documentos existentes por cada colección (entidad).

3.2. Requisitos No Funcionales

1. El aplicativo al momento de insertar cualquier entidad no requiere campos obligatorios.
2. El aplicativo debe mantener los datos almacenados seguros.
3. El aplicativo cuenta con una interfaz con botones de color azul y azul claro para las opciones que están habilitadas y deshabilitadas respectivamente, ya que puede ocurrir que determinado usuario no cuente con los permisos para realizar dicha acción.
4. Para su correcto funcionamiento, además de tener acceso a internet deben instalarse los siguientes programas: NodeJs, un navegador que soporte Javascript, MongoDB.
5. El esquema preconceptual debe ser realizado en draw.io.

4. Ejemplo Funcional

Como ya se ha mencionado anteriormente, para demostrar la funcionalidad de la aplicación se propone el modelamiento de un menú de restaurante incluyendo los datos principales del mismo con base en un esquema preconceptual y utilizando las reglas mencionadas anteriormente.

Como es común en un menú, aparecen las propiedades que tiene el restaurante como Nombre, Descripción, Ubicación, Teléfonos de Contacto y Email. Un actor “Administrador” tendrá la posibilidad de Insertar, Ver, Borrar y Actualizar la comida, entidad que agrupa tanto platos como bebidas, con los siguientes atributos: Categoría, Nombre, Precio, Descripción y

Foto. El actor “Invitado” tiene únicamente la posibilidad de Ver tanto la información del restaurante como los platos y las bebidas ofrecidas, con los atributos antes descritos. Esta es una situación común al ser un comensal en un restaurante. El esquema resultante se muestra en la siguiente figura.

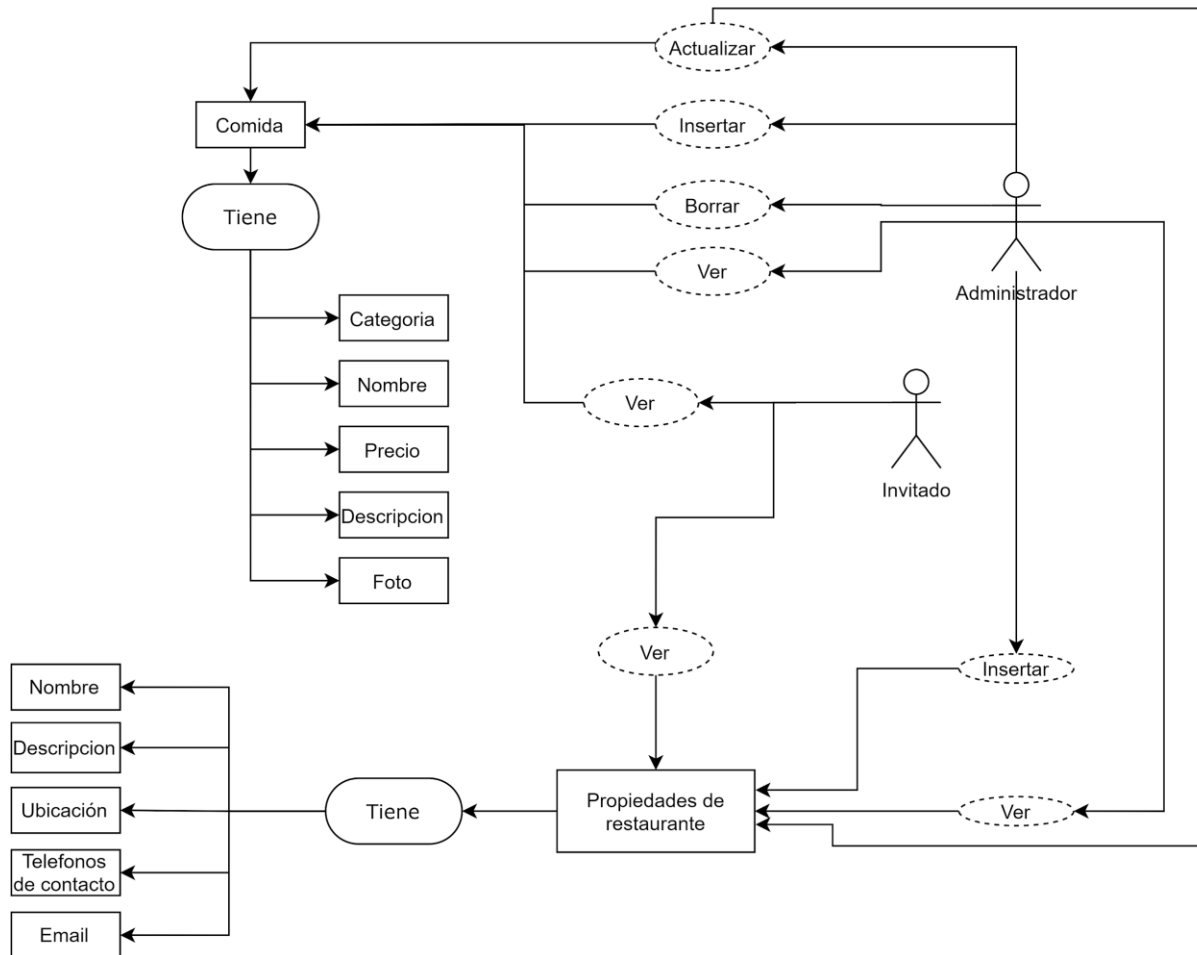


Figura 6. Esquema Preconceptual Ejemplo Restaurante.

Luego de descargar el esquema en el formato adecuado, se procede a abrir el aplicativo. La pantalla inicial corresponde a aquella mostrada en la Figura 7, y el botón “Seleccionar Archivo” se utiliza para subir el archivo correspondiente al esquema descargado.

Luego de cargar el esquema preconceptual, aparece el menú principal con los diferentes tipos de usuarios que hallan sido designados y el respectivo link para compartir la aplicación, como se muestra en la Figura 8. Se pueden utilizar los botones “Entrar” de cada caso para ingresar a la interfaz de cada uno.

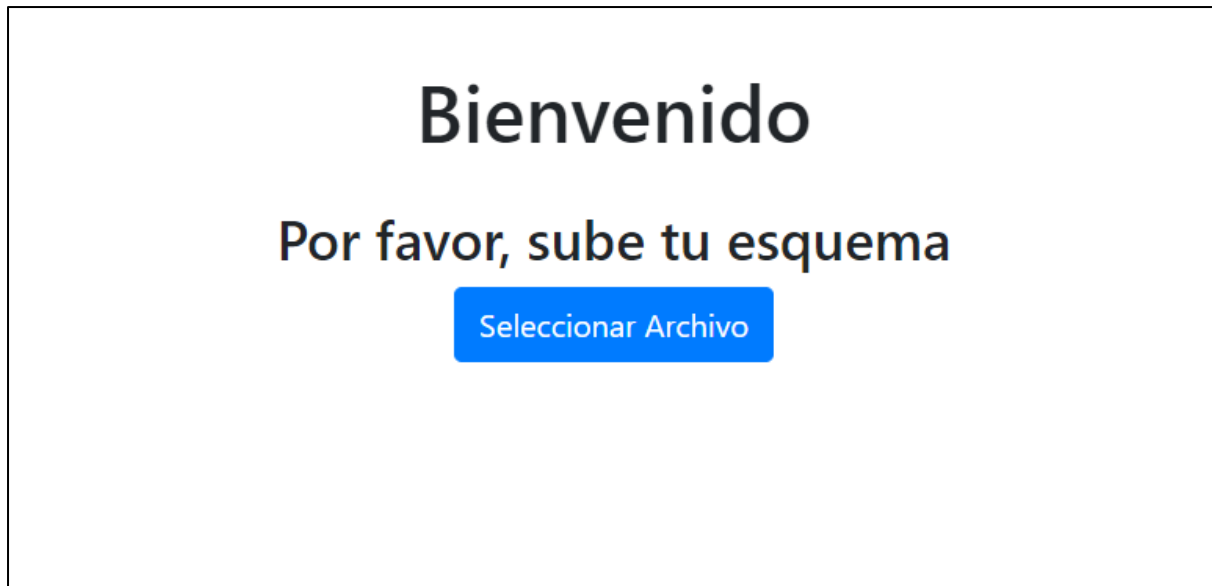


Figura 7. Menú Inicio Cargar archivo

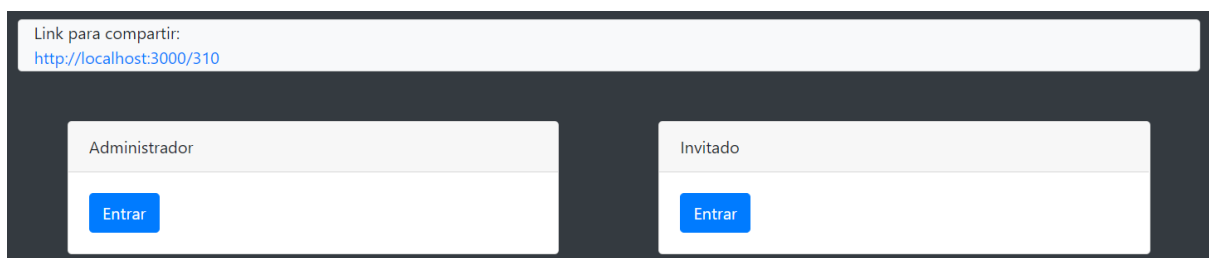


Figura 8. Menú Principal MenúApp.

Al acceder como **Administrador** aparece el siguiente menú:

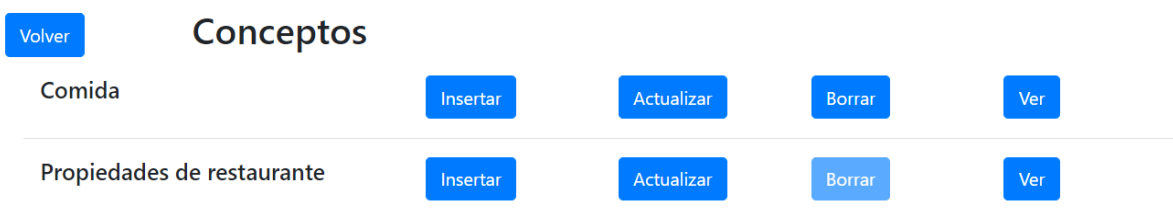


Figura 9. Menú del Administrador.

Para este ejemplo el administrador puede realizar casi todas las acciones de cada concepto, excepto que no se pueden borrar las Propiedades de restaurante ya que en el esquema preconceptual no se habilitó esta opción.

Al dar click en Insertar de Comida nos aparece el siguiente formulario.

Formulario Comida

Terminar

Precio

Nombre

Descripcion

Foto

Categoría

Guardar Informacion

Figura 10. Formulario insertar Comida.

Se llenan los datos de la nueva comida llamada “hamburguesa”, la cual pertenece a la categoría Comidas Rápidas, se da click en Guardar Información y aparece el mensaje de Guardado como se muestra a continuación.

Formulario Comida

Terminar

Precio

20000

Nombre

Hamburguesa

Descripcion

Viene con doble carne, y queso cheddar

Foto

https://t2.rg.ltmcdn.com/es/images/0/8/9/img_pan_para_hamburguesa_28980_orig.jpg

Categoría

Comidas Rápidas

Guardar Informacion

Guardado

Figura 11. Formulario insertar Comida completado y guardado (hamburguesa).

Se da click en Terminar para volver al Menú Administrador (Figura 4) para poder insertar otra comida.

Formulario Comida

Terminar

Precio

4000

Nombre

Perro Caliente

Descripcion

Viene con queso y cebolla

Foto

https://cdn.shopify.com/s/files/1/1044/8640/articles/07292017_01_portada_1024x1024.png?v=1502731526

Categoría

Comidas Rápidas

Guardar Informacion

Guardado

Figura 12. Formulario insertar Comida completado y guardado (Perro Caliente).

Se creó una nueva comida, Perro Caliente, la cual pertenece a la categoría Comidas Rápidas y sus respectivos datos.

Se da click en Terminar para volver al Menú Administrador (Figura 9), y posteriormente en el botón Ver de Comida donde aparece el listado de comidas recién creadas como se muestra a continuación.

Valores en concepto Comida

Precio: 20000 Nombre: Hamburguesa Descripción: Viene con doble carne y queso cheddar Foto: https://t2.rg.ltmcdn.com/es/images/0/8/9/img_pan_para_hamburguesa_28980_orig.jpg Categoría: Comidas Rápidas	Precio: 4000 Nombre: Perro Caliente Descripción: Viene con queso y cebolla Foto: https://cdn.shopify.com/s/files/1/1044/8640/articles/07292017_01_portada_1024x1024.png?v=1502731526 Categoría: Comidas Rápidas
---	--

Volver

Figura 13. Ver Comidas.

Al dar click en el botón Actualizar de Comida aparece el listado de todas las comidas creadas y se selecciona la que se desea modificar (resaltado en color naranja). Posteriormente se da click en seleccionar y esto nos lleva a la interfaz de la Figura 15.

Volver	Seleccionar un Concepto	Volver	Seleccionar un Concepto
Precio: 20000 Nombre: Hamburguesa Descripción: Viene con doble carne y queso cheddar Foto: https://t2.rg.ltmcdn.com/es/images/0/8/9/img_pan_para_hamburguesa_28980_orig.jpg Categoría: Comidas Rápidas	Precio: 4000 Nombre: Perro Caliente Descripción: Viene con queso y cebolla Foto: https://cdn.shopify.com/s/files/1/1044/8640/articles/07292017_01_portada_1024x1024.png?v=1502731526 Categoría: Comidas Rápidas	Precio: 20000 Nombre: Hamburguesa Descripción: Viene con doble carne y queso cheddar Foto: https://t2.rg.ltmcdn.com/es/images/0/8/9/img_pan_para_hamburguesa_28980_orig.jpg Categoría: Comidas Rápidas	Precio: 4000 Nombre: Perro Caliente Descripción: Viene con queso y cebolla Foto: https://cdn.shopify.com/s/files/1/1044/8640/articles/07292017_01_portada_1024x1024.png?v=1502731526 Categoría: Comidas Rápidas
Seleccionar		Seleccionar	

Figura 14. Actualizar Comida. Antes y después de seleccionar.

Formulario Comida
Terminar

Precio 25000

Nombre

Descripción

Foto

Categoría

Guardar Información

Guardado

Figura 15. Formulario para actualizar Comida (Hamburguesa).

Se llenan los datos que se quieren modificar, en este caso solo se quiere cambiar el precio de la hamburguesa de 20000 a 25000 y se da click en Guardar Información. Posteriormente se da click en Terminar para volver al Menú Administrador (Figura 9) y luego en el botón Ver de Comida para observar que los cambios fueron realizados correctamente.

Valores en concepto Comida

Precio: 25000 Nombre: Hamburguesa Descripcion: Viene con doble carne y queso cheddar Foto: https://t2.rg.lt/mcdn.com/es/image/s/0/8/9/img_pan_para_hamburguesa_28980_orig.jpg Categoria: Comidas Rápidas	Precio: 4000 Nombre: Perro Caliente Descripcion: Viene con queso y cebolla Foto: https://cdn.shopify.com/s/files/1/1044/8640/articles/07292017_01_portada_1024x1024.png?v=1502731526 Categoria: Comidas Rápidas
---	--

[Volver](#)

Figura 16. Ver Comidas Actualizado.

Al dar click en el botón Insertar de Propiedades de restaurante nos aparece en siguiente formulario.

Formulario Propiedades de restaurante
[Terminar](#)

Nombre	La Hamburguesia
Descripcion	Las mejores hamburguesas
Ubicación	El Poblado
Telefonos de contacto	4228919
Email	hamburguesia@gmail.com

[Guardar Informacion](#)
Guardado

Figura 17. Formulario insertar Propiedades de restaurante.

Se da click en el botón Guardar información y posteriormente en el botón Terminar para volver al Menú Administrador (Figura 9) y luego en el botón Ver de Propiedades de restaurante para observar que los datos fueron realizados correctamente como se muestra en la siguiente figura.

Valores en concepto Propiedades de restaurante

Nombre:La
Hamburguesia
Descripción:Las
mejores
hamburguesas
Ubicación:El
Poblado
**Telefonos de
contacto:**4228919
Email:hamburguesia
@gmail.com

Volver

Figura 18. Ver Propiedades de restaurante.

Al acceder como **Invitado** tenemos el siguiente menú:

Volver	Conceptos				
Comida	Insertar	Actualizar	Borrar	Ver	
Propiedades de restaurante	Insertar	Actualizar	Borrar	Ver	

Figura 19. Menú Invitado

Se puede observar que solo tiene acceso al Ver de Comida y Propiedades de restaurante, al dar click en cada uno de ellos, aparecerá el mismo contenido mostrado anteriormente en la Figura 16 y Figura 18.

5. Conclusiones

- La creación y desarrollo de una aplicación CASE como la realizada en este trabajo, permitió identificar cada fase del proceso de compilación enseñado en clase. Ejemplificando diferentes conceptos, y mostrando lo rigurosa y detallada que debe ser cada fase para poder garantizar una transformación adecuada de un esquema visual en una aplicación funcional.
- Es de vital importancia formular correctamente las reglas para la creación del esquema preconceptual, aquellas para traducirlo a un archivo .xml, luego a formato

json y aquellas que determinan el uso funcional de la aplicación, con el fin de realizar un producto de trabajo eficiente y con todos los requerimientos establecidos.

- Dada la flexibilidad establecida para añadir, eliminar o modificar las reglas fue posible hacer ajustes con relativa facilidad. En casos de la vida real en donde el cliente puede añadir o cambiar de opinión de un día para otro, es fundamental que la aplicación sea dinámica a la hora de recibir ajustes.
- El desarrollo enfocado a la identificación de componentes y valores dinámicos que se deciden en momento de procesamiento y parseo se ve beneficiado por una arquitectura modular con responsabilidades definidas e interfaces amigables que hacen provecho de las propiedades que se reflejan en las buenas prácticas del desarrollo de software como la encapsulación, la abstracción de responsabilidades y la subdivisión de tareas.
- El uso de aplicativos dinámicos basados en diagramas es una herramienta poderosa que facilita al usuario final el manejo de tareas repetitivas. MenúApp refleja la formulación, construcción y distribución de un formulario, basado en principios de autorización a característica del aplicativo usando listas de permisos. Sin embargo, gracias a la flexibilidad que se maneja con este aplicativo, podría ser expandido a otros problemas encontrados frecuentemente en el desarrollo de soluciones informáticas, como son por ejemplo la generación de sistemas gestores de inventario, pedido de productos a domicilio, generador de facturas, entre otros.

6. Referencias

Object Oriented Software Engineering. Bernd Bruegge y Allen H.Dutoit. Prentice Hall, 2000.

Software Requirements. Karl. E.Wiegers. Microsoft Press, 1999.

ZAPATA C.M y LEZCANO L. Caracterización de los verbos usados en el diagrama de objetivos. DYNA, Vol. 76, Núm. 158 (2009)