

INSTITUTO TECNOLÓGICO AUTÓNOMO DE
MÉXICO



DESARROLLO DE APLICACIONES
INFORMÁTICAS

PROYECTO 1: Tamalito



Integrantes:

Ángel Zatarain López 171880

José Carlos López 165040

Diego Hernández Delgado 176262

Grupo: 002

Maestra: Ana Lidia Franzoni

Fecha de entrega: 05-noviembre-2019

Índice

1. INTRODUCCIÓN	2
1.1 Contexto	
1.2 Identificación del Problema	
1.3 Objetivo(s)	
2. ANÁLISIS	2
2.1 Requerimientos funcionales	
2.2 Restricciones	
2.3 Trabajos Relacionados	
3. DISEÑO	3
3.1 Soluciones Alternativas	
3.2 Descripción completa del diseño (entidad-vínculo, relacional y script SQL)	
3.3 Estándares Utilizados	
4. IMPLEMENTACIÓN	6
4.1 Especificaciones de Software y Hardware	
4.2 Descripción completa de la implementación	
5. PRUEBAS Y RESULTADOS	7
5.1 Descripción de las pruebas	
5.2 Resultados	
5.3 Análisis de resultados	
6. CONCLUSIONES	8
6.1 Impacto ético, global, económico, social y ambiental	
6.2 Conclusiones	
7. REFERENCIAS	8
8. MANUAL DE USUARIO	9
8.1 WPF	
8.2 WEB	

INTRODUCCIÓN

Contexto:

Nuestro proyecto ha sido contextualizado para el uso diario tanto de los estudiantes, como de los empleados y profesores del Instituto Tecnológico Autónomo de México (ITAM).

Identificación del problema:

Se ha notado un disgusto general por la nueva cafetería de nuestra H. Institución, debido a sus elevados precios.

Objetivos:

La intención de nuestro proyecto, *Tamalito*, es brindarle tanto a estudiantes como a maestro y empleados una alternativa gastronómica a la cafetería del ITAM. La solución consiste en un negocio de cocina mexicana tradicional, sirviendo tanto tamales como atoles en una variedad de sabores.

ANÁLISIS

Requerimientos funcionales:

El proyecto se divide en dos partes: WPF y WEB. Cada una de dichas partes debe desempeñar una funcionalidad diferente, ya que una está más orientada al uso por parte de aquellos que controlan el negocio (WPF) y la otra está orientada hacia la interacción entre el consumidor y el negocio (WEB). Habiendo establecido esta diferencia podemos decir que el WPF debe ser capaz de: controlar el acceso a diferentes métodos dependiendo del tipo de usuario (dueño, gerente o empleado); permitirle a cualquier tipo de usuario el tomar una orden; permitirle únicamente al dueño y los gerentes dar de alta, de baja, o promover a un empleado; darle la capacidad, tanto al dueño como a los gerentes, de añadir inventario; por último, debe poder darle acceso al dueño a la información de las ventas y otorgarle el permiso para alterar los costos de los productos.

La segunda parte del proyecto (WEB) debe mediar la comunicación entre usuario y negocio. Para esto primero debe tener una página de presentación con información sobre el negocio; un tipo de carta de presentación del negocio. Igualmente, debe permitirle al usuario registrarse en la base de datos; esto para tener un registro de quienes compran nuestros productos. Tras haberse registrado, el usuario debe poder ingresar al programa para hacer pedidos. Un usuario puede hacer más de un pedido y en cada pedido puede pedir cuantos productos quiera. Finalmente, el usuario debe ser capaz de cerrar sus sesión de *Tamalito*.

Ambas partes del programa, WPF y WEB, deben mostrar una interfaz gráfica amigable con su respectivo usuario.

Restricciones:

La restricción de implementación es que tiene que ser ejecutado con Visual Studio 2019 y con SQL Server.

Algunas de las restricciones de uso de la aplicación WEB son que un usuario no puede registrar más de una cuenta por cada correo electrónico, y al momento de realizar una orden debe presionar dos botones, el primero para actualizar su orden y el segundo para confirmar la misma. En el caso de la aplicación de escritorio, la restricción principal es que no todos los tipos de usuarios pueden acceder a todos los métodos existentes; es decir, el dueño puede acceder a todos los métodos, el gerente a menos que el dueño y el empleado común únicamente puede tomar los pedidos de los clientes. Esto con el fin de mantener un control claro en el negocio.

Trabajos relacionados:

Tamalito puede identificarse con cualquier otro sistema en el cual un usuario puede comunicarse, haciendo pedidos desde una página web, con un restaurante desde una ubicación remota. Ejemplos de esto pueden ser Dominoes, UberEats y Rappi. Esto por la parte de la aplicación Web. Por el lado de la aplicación WPF, el proyecto puede identificarse con cualquier otro sistema de control centralizado de un restaurante.

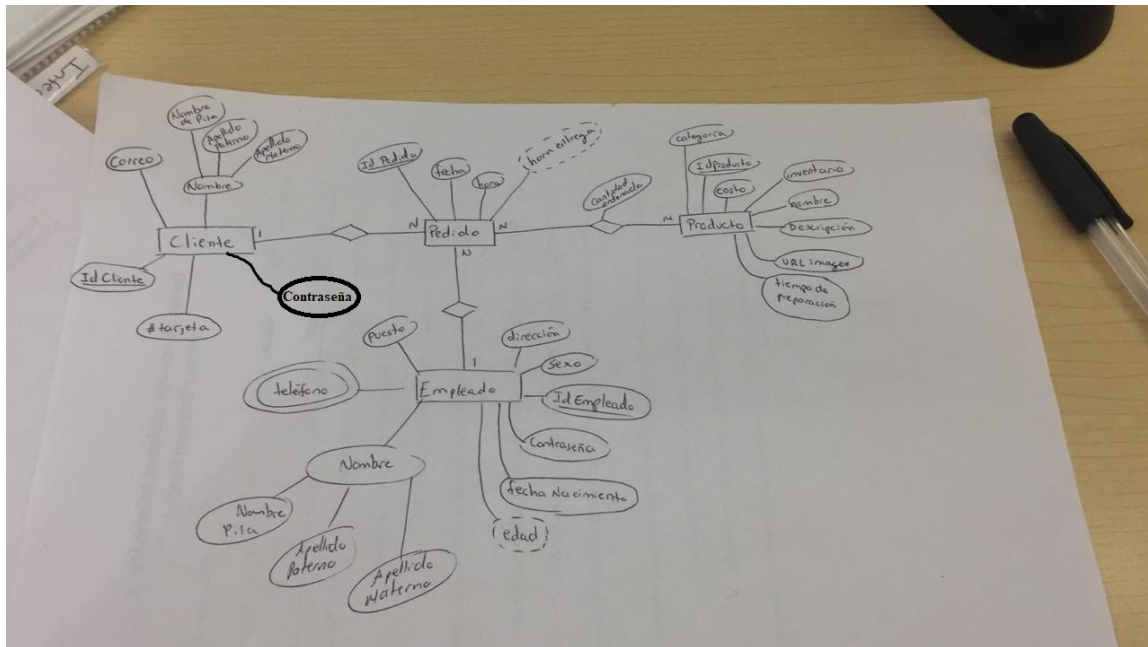
DISEÑO

Soluciones alternativas:

Una solución alternativa es el desarrollo de aplicaciones móviles nativas para los sistemas operativos Android y IOS.

Descripción completa del diseño:

MODELO ENTIDAD-VÍNCULO



MODELO RELACIONAL

Empleado (idEmpleado, nombre, apellidoP, apellidoM, fechaNac, sexo, dirección, puesto, contraseña, activo)

Teléfono (idEmpleado, teléfono)

Pedido (idPedido, fecha, idCliente(FK), idEmpleado(FK))

Cliente (idCliente, numTarjeta, correo, nombre, apellidoP, apellidoM, contra)

Ped_Prod (idPedido(FK), idProducto(FK), cantidad)

Producto (idProd, costo, inventario, nombre, descripción, categoría, tiempoPreparación, urlImagen)

SCRIPT - SQL

```
CREATE DATABASE tamalito;
```

```
USE tamalito;
```

```
CREATE TABLE empleados(
```

```
idEmpleado INTEGER IDENTITY(1,1) PRIMARY KEY ,
```

```
nombre VARCHAR(30) NOT NULL,
```

```
apellidoP VARCHAR(30) NOT NULL,
```

```

apellidoM VARCHAR(30) NOT NULL,

fechaNac DATE,

sexo CHAR(1) NOT NULL,

direccion VARCHAR(40),

puesto VARCHAR(20) NOT NULL,

contrasenia VARCHAR(16) NOT NULL,

activo TINYINT DEFAULT 1

);

CREATE TABLE telefonos(

idEmpleado INTEGER PRIMARY KEY,

telefono INTEGER,

);

CREATE TABLE clientes(

idCliente INTEGER IDENTITY(1,1) PRIMARY KEY,

numTarjeta VARCHAR(30),

correo VARCHAR(30),

nombre VARCHAR(30),

apellidoP VARCHAR(30),

apellidoM VARCHAR(30),

contra VARCHAR(30) NOT NULL

);

CREATE TABLE pedidos(

idPedido INTEGER IDENTITY(1,1) PRIMARY KEY,

fechaHora DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,

idCliente INTEGER REFERENCES clientes(idCliente),

idEmpleado INTEGER REFERENCES empleados(idEmpleado)

);

CREATE TABLE productos(

idProducto INTEGER IDENTITY(1,1) PRIMARY KEY,

costo INTEGER,

```

```

inventario INTEGER, --cantidad de ese producto

nombre VARCHAR(30),

descripcion VARCHAR(270),

categoria VARCHAR(30),

tiempoPreparacion INTEGER,

urlImagen VARCHAR(50)

);

CREATE TABLE pedidosProductos(

idPedido INTEGER FOREIGN KEY REFERENCES pedidos(idPedido),

idProducto INTEGER FOREIGN KEY REFERENCES productos(idProducto),

cantidad INTEGER --cantidad de ese producto

);

-----INITIALIZE
VALUES-----

--EMPLEADOS--

INSERT INTO empleados(nombre, apellidoP, apellidoM, fechaNac, sexo, direccion, puesto,
contrasenia,activo)

VALUES('Angel', 'Zatarain', 'López', '1999-03-11', 'H', 'Cancún', 'Empleado', '1', 1);

INSERT INTO empleados( nombre, apellidoP, apellidoM, fechaNac, sexo, direccion,
puesto, contrasenia,activo)

VALUES('José', 'Cortés', 'Gasca', '2019-12-10', 'H', 'CDMX', 'Gerente', '2', 1);

INSERT INTO empleados(nombre, apellidoP, apellidoM, fechaNac, sexo, direccion, puesto,
contrasenia, activo)

VALUES('Diego', 'Hernández', 'Delgado', '2019-11-10', 'H', 'EUA', 'Dueño', '3',1);

INSERT INTO telefonos(idEmpleado, telefono) VALUES(1, 55543267);

INSERT INTO telefonos(idEmpleado, telefono) VALUES(2, 55549999);

```

```
INSERT INTO telefonos(idEmpleado, telefono) VALUES(3, 55567888);
```

```
--CLIENTES--
```

```
INSERT INTO clientes(numTarjeta, correo, nombre, apellidoP, apellidoM, contra) VALUES(
'55658932', 'cliente1@gmail.com', 'Juan', 'Gómez', 'Peralta', '1');
```

```
INSERT INTO clientes(numTarjeta, correo, nombre, apellidoP, apellidoM, contra) VALUES(
'55677733', 'cliente2@gmail.com', 'Paco', 'Pérez', 'Lenovos', '2');
```

```
INSERT INTO clientes(numTarjeta, correo, nombre, apellidoP, apellidoM, contra) VALUES(
'55999988', 'cliente3@gmail.com', 'Luis', 'Casas', 'Bonitas', '3');
```

```
--PRODUCTOS--
```

```
--TAMALES--
```

```
INSERT INTO productos(costo, inventario, nombre, descripcion, categoria,
tiempoPreparacion, urlImagen)
```

```
VALUES(15, 20, 'verde', 'Delicioso tamal verde', 'tamal', 30,
'./images/productos/tamales/verde.jpg');
```

```
INSERT INTO productos(costo, inventario, nombre, descripcion, categoria,
tiempoPreparacion, urlImagen)
```

```
VALUES(15, 20, 'rojo', 'Delicioso tamal rojo', 'tamal', 30,
'./images/productos/tamales/rojo.jpg');
```

```
INSERT INTO productos(costo, inventario, nombre, descripcion, categoria,
tiempoPreparacion, urlImagen)
```

```
VALUES(15, 20, 'mole', 'Delicioso tamal mole', 'tamal', 30,
'./images/productos/tamales/mole.jpg');
```

```
INSERT INTO productos(costo, inventario, nombre, descripcion, categoria,
tiempoPreparacion, urlImagen)
```

```
VALUES(15, 20, 'dulce', 'Delicioso tamal dulce', 'tamal', 30,
'./images/productos/tamales/dulce.jpg');
```

```
--ATOLES--
```

```
INSERT INTO productos(costo, inventario, nombre, descripcion, categoria,
tiempoPreparacion, urlImagen)
```



```
VALUES(12, 30, 'arroz', 'Delicioso atole de arroz', 'atole', 20,
'./images/productos/atoles/verde.jpg');
```

```
INSERT INTO productos(costo, inventario, nombre, descripcion, categoria,
tiempoPreparacion, urlImagen)
```

```
VALUES(12, 30, 'vainilla', 'Delicioso atole de vainilla', 'atole', 20,
'./images/productos/atoles/rojo.jpg');
```

```
INSERT INTO productos(costo, inventario, nombre, descripcion, categoria,
tiempoPreparacion, urlImagen)
```

```
VALUES(12, 30, 'fresa', 'Delicioso atole de fresa', 'atole', 20,
'./images/productos/atoles/mole.jpg');
```

```
INSERT INTO productos(costo, inventario, nombre, descripcion, categoria,
tiempoPreparacion, urlImagen)
```

```
VALUES(12, 30, 'chocolate', 'Delicioso atole de chocolate', 'atole', 20,
'./images/productos/atoles/dulce.jpg');
```

```
--PEDIDOS--
```

```
INSERT INTO pedidos(idCliente, idEmpleado) VALUES(1,1);
```

```
INSERT INTO pedidos(idCliente, idEmpleado) VALUES(2,2);
```

```
INSERT INTO pedidos(idCliente, idEmpleado) VALUES(3,3);
```

```
INSERT INTO pedidosProductos(idPedido, idProducto, cantidad) VALUES(1,1,2);
```

```
INSERT INTO pedidosProductos(idPedido, idProducto, cantidad) VALUES(1,3,2);
```

```
INSERT INTO pedidosProductos(idPedido, idProducto, cantidad) VALUES(1,4,5);
```

```
INSERT INTO pedidosProductos(idPedido, idProducto, cantidad) VALUES(2,1,1);
```

```
INSERT INTO pedidosProductos(idPedido, idProducto, cantidad) VALUES(2,2,1);
```

```
INSERT INTO pedidosProductos(idPedido, idProducto, cantidad) VALUES(3,2,3);
```

```
INSERT INTO pedidosProductos(idPedido, idProducto, cantidad) VALUES(3,3,3);
```

```
INSERT INTO pedidosProductos(idPedido, idProducto, cantidad) VALUES(3,4,3);
```

Estándares Utilizados:

Para organización de las labores en los proyectos se utilizó la metodología de trabajo Scrum donde se dividieron las tareas por objetivos a lo largo de tres semanas. Para el diseño de la base de datos se utilizaron los modelos entidad-vínculo y relacional. Para el diseño de la aplicación de escritorio (WPF) se implementaron las librerías “MaterialDesign” y “MaterialColorDesign”, mientras que el funcionamiento se desarrolló en el lenguaje de programación C#. Para la aplicación web las vistas se estructuraron con HTML, el diseño y las animaciones con CSS, y el dinamismo con C#.

IMPLEMENTACIÓN

Especificaciones de software y hardware:

En cuanto a especificaciones de software, el programa debe implementarse con Visual Studio 2019 y SQL Server. En cuanto a hardware, con una computadora con una cantidad regular de RAM será suficiente.

Descripción completa de la implementación:

Primero se debe inicializar la base de datos con el script proporcionado (ver sección 3.2) en SQL Server. Después de esto se debe abrir el proyecto de WPF y WEB en Visual Studio 2019. Antes de poder correr el proyecto debemos revisar la conexión, solo para comprobar que está correcta. Una vez que nuestras conexiones estén correctas ya podemos inicializar los proyectos, tanto WPF como WEB.

PRUEBAS Y RESULTADOS

Descripción de las pruebas:

Para poder asegurarnos de que el proyecto funcionará correctamente tuvimos que hacer una serie de pruebas tanto en el WPF como en el WEB. Para el WPF la primera prueba fue ver que solo le diera acceso a aquellos usuarios registrados, que estuviesen empleados en ese momento, y que a cada usuario lo llevase a su respectiva ventana. El botón de tomar orden debía permitirle a cualquier empleado tomar una orden y esta orden tenía que registrarse en la base de datos. Los métodos de dar de alta un empleado, dar de baja a un empleado y promover a un empleado debían registrar a un empleado en la base de datos, modificar la “actividad” de un empleado en la base de datos (de activo a inactivo), y modificar el puesto de un empleado en la base de datos respectivamente. Por último, debía dejar que el dueño consultara las ventas de un periodo de tiempo, mandando un mensaje en

caso de haber un error en las fechas (que la fecha inicial sea, cronológicamente, más reciente que la fecha final), y dejarle modificar el costo de los productos. Igualmente, cuando un empleado no estaba registrado, o había sido dado de baja, el programa no le podía dar acceso.

Para la aplicación WEB las pruebas consistieron en dar de alta a un usuario, ingresar con un usuario y generar una orden.

Resultados:

Los resultados de nuestras pruebas mostraron que en el WPF, a menos de tener un usuario que estuviese activo, poniendo su contraseña correctamente, no se podría entrar al programa. Igualmente, se hizo lo esperado con respecto a la restricción de los usuarios en cuanto a qué métodos podrían acceder, otorgando la mayor capacidad al dueño, seguido del gerente y el empleado común. Los métodos programados funcionaron de forma debida; es decir, el “alta empleado” si registraba un nuevo empleado en la base de datos, el “baja empleado” desactivaba al empleado correspondiente de la base de datos, el “modificar puesto” si modifica el puesto del empleado correspondiente en la base de datos (evitando que un empleado modificara su propio puesto), el método de “añadir inventarios” si permite añadir unidades de inventario a la base de dato, “consultar ventas” si mostraba una lista de todas las ventas efectuadas en un periodo de tiempo (siempre y cuando las fechas fuesen válidas), por último, el botón “modificar costos” igual permitía modificar el costo de algún producto y alteraba el mismo en la base de datos.

Con respecto al WEB, al momento de registrarse, un usuario no podía tener más de una cuenta por correo electrónico y si trataba de registrarse más de una vez con un mismo correo se le mostraba un mensaje advirtiéndolo que no podía hacerlo. En caso de registrarse correctamente (un correo por cuenta), el usuario se registraba en la base de datos. Solo se puede entrar al programa con un usuario y contraseña válidos y correctos. Al momento de tomar una orden primero se debe presionar el botón de “ver carrito” para actualizar el carrito del consumidor y luego se debía presionar el botón de confirmar compra, este botón registra la compra en la base de datos. Finalmente, al presionar la etiqueta “Log-Out” el usuario cerraba su sesión.

Análisis de resultados:

Los resultados nos mostraron que ambos programas tienen un estricto protocolo de ingreso de usuario, al igual que ser capaz de controlar el acceso a determinados métodos en ambos casos.

CONCLUSIONES

Impacto ético, global, económico, social y ambiental:

El impacto ético de nuestro proyecto es que logrará fomentar la sana competencia en el mercado. El impacto global será que la humanidad, en conjunto, pueda disfrutar más de la

gastronomía tradicional mexicana. Al promover el consumo, el proyecto logrará impulsar positivamente al PIB y al crecimiento de la economía mexicana. En *Tamalito* con cada compra ayudas a las tortugas, a través de donaciones indirectas, por lo que con esto se puede notar el impacto ambiental de nuestro proyecto. Finalmente, el impacto social será desarrollar una mayor apreciación por la cocina tradicional mexicana en nuestra comunidad.

Conclusiones:

El proyecto es amigable con el usuario, tanto del lado del WPF como del WEB, siendo lo más claro posible para evitar complicaciones. Igualmente, *Tamalito* puede provocar un cambio positivo en la sociedad, tanto de forma ética, como económica y ambiental, por lo que es un proyecto en el que vale la pena invertir.

REFERENCIAS

1. F. López, Notas del curso de Desarrollo de Aplicaciones Informáticas, ITAM.
2. Raffaele Garofalo, Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern, Microsoft Press, ISBN: 978-0-73565092-3, 2011.
3. I. Spaanjaars, Beginning ASP.NET 4.5.1 in C# and VB, John Wiley & Sons, 2014.
4. Documentación de C# y Visual Studio
5. Consultas a Stackoverflow y w3schools

MANUALES DE USUARIO:

WPF:

Tamalito

Manual de Usuario

WPF



Al empezar la aplicación Tamalito, la primera pantalla dará la bienvenida al usuario.

El botón 'Comenzar' dará paso al inicio de sesión.

A continuación, se deberá iniciar sesión ingresando el Usuario y la respectiva contraseña.

Si el usuario y la contraseña son correctas, el botón 'Iniciar Sesión' dará paso a la pantalla ya sea de Dueño, Gerente o Empleado, dependiendo del puesto del usuario en turno.

The image shows a mobile application interface for 'Tamalito'. At the top, there is a dark blue header with the word 'Tamalito' in white. Below the header, the background is light gray. There are two input fields: the first is labeled 'Usuario' with a person icon, and the second is labeled 'Contraseña' with a key icon. Both fields have horizontal lines indicating where to enter text. At the bottom center, there is a blue button with the text 'Iniciar sesión' in white. The entire interface is framed by a blue border, and there are navigation icons (back and close) in the top left and right corners of the app window.

Usuarios

Empleado

Un usuario Empleado únicamente puede tomar orden al cliente.

Al igual que para un Dueño o un Gerente, el botón de 'Cerrar Sesión' manda al usuario de regreso a la ventana de Inicio de Sesión



Gerente

Además de poder tomar orden, la ventana de un usuario 'Gerente' incluye botones que permiten añadir elementos de un producto al inventario, dar de alta un empleado, cambiar el puesto de un empleado y, por último, dar de baja a un trabajador.

Dueño

Si se trata de un usuario 'Dueño', éste tiene todos los botones de un usuario Gerente, además de un botón que le permite modificar los precios de los productos y otro que donde consultará las ventas de cierto periodo de tiempo.



Modificar Precio

Esta es una facultad exclusiva de los ‘Dueños’. El usuario debe ingresar el número y nombre del producto cuyo precio desea modificar.



The screenshot shows a mobile application window titled 'Modificar precio'. It contains four input fields: 'Id Producto', 'Nombre', 'Precio Actual', and 'Nuevo Precio'. Below the 'Nombre' field is a blue button labeled 'Consultar'. Below the 'Nuevo Precio' field is a blue button labeled 'Modificar'.



This screenshot shows the same 'Modificar precio' form after the 'Consultar' button has been pressed. The 'Id Producto' field now contains the number '4' and the 'Nombre' field contains the word 'dulce'. The 'Precio Actual' field now displays the value '15'. The 'Nuevo Precio' field is still empty. The 'Consultar' and 'Modificar' buttons remain visible.

Al presionar el botón ‘Consultar’, se mostrará el precio actual del producto ingresado.

A continuación, se puede ingresar el nuevo precio deseado.

Después de presionar el botón ‘Modificar’, el programa mostrará un mensaje de éxito.



This screenshot shows the 'Modificar precio' form with 'Id Producto' as '4' and 'Nombre' as 'dulce'. The 'Precio Actual' is '15' and the 'Nuevo Precio' is '20'. A small dialog box is overlaid on the form, displaying the message 'Precio actualizado.' with an 'Aceptar' button. The 'Modificar' button is still visible at the bottom.

Consultar Ventas

Sólo los ‘Dueños’ pueden hacer una consulta de ventas.

Consiste en seleccionar una fecha inicial, una fecha final y finalmente oprimir el botón ‘Consultar’

The screenshot shows a window titled 'Consulta de ventas'. It contains two rows of date pickers. The first row is for 'Fecha inicio' with fields for 'Año', 'Mes', and 'Día'. The second row is for 'Fecha final' with similar fields. A 'Consultar' button is located to the right of the date pickers. Below the date pickers is a large empty space, and at the bottom, there is a 'Total:' label followed by a text input field.

The screenshot shows the same 'Consulta de ventas' window, but now it displays a table of sales results. The table has four columns: 'numPed', 'numProd', 'fechaProd', and 'ganancia'. Below the table, the 'Total:' label is followed by the value '\$ 300'.

numPed	numProd	fechaProd	ganancia
1	1	11/4/2019 6:04:25 PM	30
1	3	11/4/2019 6:04:25 PM	30
1	4	11/4/2019 6:04:25 PM	75
2	1	11/4/2019 6:04:25 PM	15
2	2	11/4/2019 6:04:25 PM	15
3	2	11/4/2019 6:04:25 PM	45
3	3	11/4/2019 6:04:25 PM	45
3	4	11/4/2019 6:04:25 PM	45

Si se ingresaron fechas coherentes, se mostrará la lista con las ventas del periodo de tiempo correspondiente, así como el monto total de las mismas.

En caso de que las fechas tengan algún tipo de incoherencia, el programa mostrará un mensaje de error.

The screenshot shows the 'Consulta de ventas' window with the date pickers set to 'Año 2018', 'Mes Mayo', 'Día 5' for the start date and 'Año 2017', 'Mes Octubre', 'Día 9' for the end date. An error dialog box is displayed in the center of the window with the text 'Las fechas están mal' and an 'Aceptar' button. The 'Total:' label and input field are visible at the bottom.

Dar de Alta un Empleado



Formulario 'Alta empleado' con los siguientes campos:

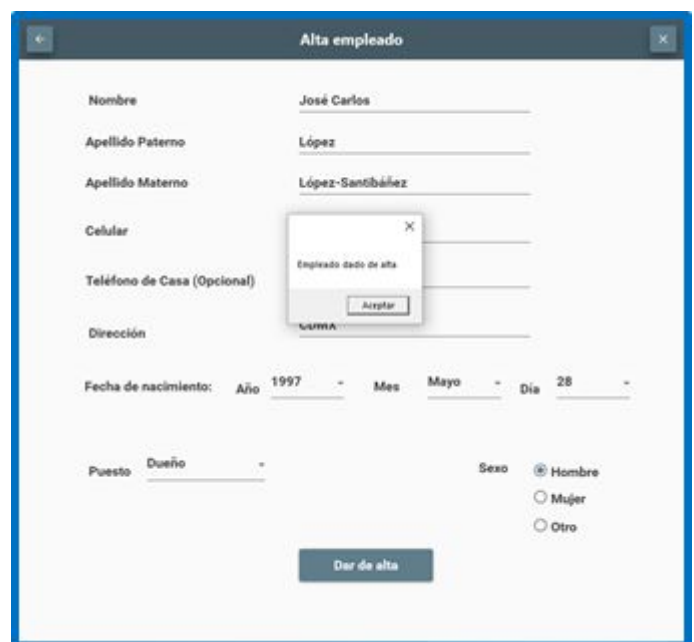
- Nombre
- Apellido Paterno
- Apellido Materno
- Celular
- Teléfono de Casa (Opcional)
- Dirección
- Fecha de nacimiento: Año, Mes, Día
- Puesto
- Sexo: ☐ Hombre, ☐ Mujer, ☐ Otro
- Botón: Dar de alta

Tanto 'Gerentes' como 'Dueños' son capaces de incorporar un nuevo trabajador a la empresa.

Para esto se debe ingresar todos los datos en sus respectivos campos.

Una vez ingresados todos los datos, el botón 'Dar de Alta' automáticamente agregará al nuevo empleado a la nómina.

El programa dará un mensaje de éxito.



Formulario 'Alta empleado' con los siguientes datos ingresados:

- Nombre: José Carlos
- Apellido Paterno: López
- Apellido Materno: López-Santibáñez
- Celular:
- Teléfono de Casa (Opcional):
- Dirección:
- Fecha de nacimiento: Año 1997, Mes Mayo, Día 28
- Puesto: Dueño
- Sexo: ☒ Hombre, ☐ Mujer, ☐ Otro
- Botón: Dar de alta

Mensaje de éxito: Empleado dado de alta. Botón: Aceptar.

Modificar Puesto



Formulario 'Modificar puesto' con los siguientes campos:

- Id Empleado
- Nombre
- Apellido Paterno
- Apellido Materno
- Nuevo puesto (con una flecha hacia abajo)

Botón 'Modificar' al fondo.

Para modificar el puesto de un usuario, es necesario primero ingresar los datos del usuario a promover o degradar.

Posteriormente, se debe presionar el botón 'Modificar'. Si los datos son correctos, se mostrará un mensaje de éxito.



Formulario 'Modificar puesto' con los siguientes datos:

- Id Empleado: 2
- Nombre: José
- Apellido: [vacío]
- Apellido: [vacío]
- Nuevo puesto: [vacío]

Se muestra un mensaje de éxito: 'se cambió el puesto del empleado'. Botón 'Aceptar' visible.



Formulario 'Modificar puesto' con los siguientes datos:

- Id Empleado: 3
- Nombre: Diego
- Apellido: [vacío]
- Apellido: [vacío]
- Nuevo puesto: [vacío]

Se muestra un mensaje de error: 'No puedes modificar tu propio puesto'. Botón 'Aceptar' visible.

Evidentemente, un usuario no debería ser capaz de darse un ascenso a sí mismo, por lo cual el programa dará un mensaje en caso de que se intente.

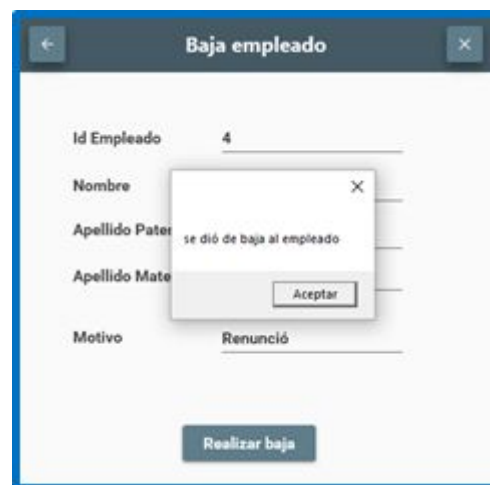
Dar de Baja un Empleado



A screenshot of a mobile application window titled 'Baja empleado'. It contains five input fields: 'Id Empleado', 'Nombre', 'Apellido Paterno', 'Apellido Materno', and 'Motivo'. At the bottom is a button labeled 'Realizar baja'.

Para dar de baja a un trabajador, el programa solicita los datos del usuario que será finiquitado, además del motivo por el cual será despedido

Al presionar el botón 'Realizar baja', el programa mostrará un mensaje de éxito.



A screenshot of the 'Baja empleado' form with the following data: Id Empleado: 4, Nombre: (empty), Apellido Paterno: (empty), Apellido Materno: (empty), Motivo: Renunció. A modal dialog box is displayed in the center with the text 'se dió de baja al empleado' and an 'Aceptar' button. The 'Realizar baja' button is at the bottom.



A screenshot of a mobile application window titled 'Tamalito'. It shows a login form with fields for 'Usuario' (containing '3') and 'Contraseña'. Below the fields is a button labeled 'Iniciar sesión'. A modal dialog box is displayed in the center with the text 'El empleado ya no trabaja aquí' and an 'Aceptar' button.

El usuario que ha sido dado de baja no podrá ingresar de nuevo a 'Tamalito', y el programa se lo hará saber al intentar iniciar sesión.

Tomar Orden



La opción de tomar un pedido la tienen todos los usuarios. La ventana muestra imágenes de todos los productos disponibles, así como sus respectivos precios.

Para agregar un producto al pedido, se debe ingresar en primer lugar la cantidad deseada y posteriormente seleccionar la imagen de dicho producto. El borde de la imagen se tonará de color verde y la orden se verá actualizada de manera automática.



Para finalizar y hacer válido el pedido, únicamente hay que apretar el botón 'Confirmar Compra', con lo que el programa mostrará un mensaje de éxito.

Añadir Cantidad de un Producto al Inventario



Añadir inventario

Id Producto

Nombre

Cantidad actual

Agrega

Para añadir unidades al inventario, el usuario debe ingresar el número del producto cuyo precio desea modificar.

Al presionar el botón ‘Consultar’, se mostrará la cantidad actual del producto ingresado.

A continuación, se puede escribir la nueva cantidad deseada.

Después de presionar el botón ‘Modificar’, el programa mostrará un mensaje de éxito.

WEB:

Tamalito

Manual de Usuario

WEB



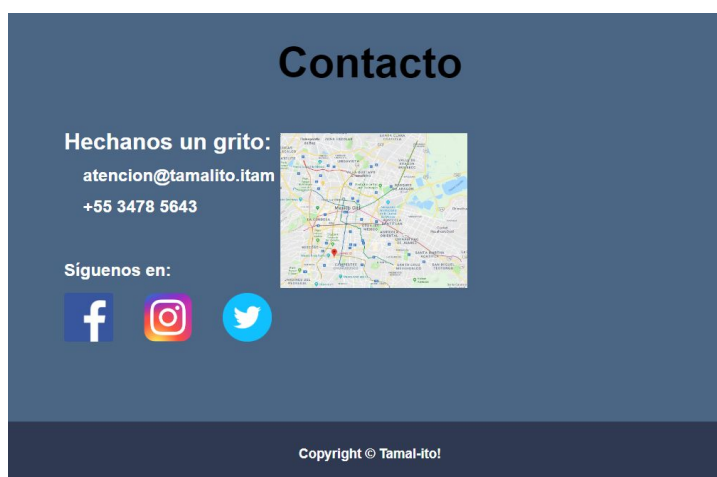
Esta es la ventana principal; es decir, el “Inicio” de la página web. Esta ventana tiene la intención de ser la tarjeta de presentación de *Tamalito*.

[Inicio](#) [Nosotros](#) [Contacto](#) [Log-In](#) [Sign-In](#)

Esta es la barra de herramientas principal de la página web. Si presionas “Inicio” te manda a la página de inicio. Si presiona “Nosotros” o “Contacto” te manda a secciones distintas de la misma página principal. Si presionas “Sign-In” llegas a una nueva página en la que podrás registrarte. Si presionas “Log-In” llegas a una nueva página en la que podrás ingresar a tu perfil.



Esta es la sección de “Nosotros” de la página inicial. Aquí se presenta una pequeña descripción del negocio e información de los encargados del mismo



Esta es la sección de “Contacto” de la página principal. Aquí se muestran las redes sociales del negocio, para que se puedan poner en contacto con nosotros.

The registration form is titled "REGISTRARSE". It contains the following fields and buttons:

- Nombre (Name)
- Apellido Paterno (Paternal Surname)
- Apellido Materno (Maternal Surname)
- Correo Electrónico (Email)
- Número de Tarjeta (Card Number)
- Password
- Enviar (Send)

Este es el menú de registro. Aquí puedes registrarte para iniciar a pedir tamales. ADVERTENCIA: No puedes registrarte más de una vez con el mismo correo electrónico.

The login form is titled "INICIAR SESIÓN". It contains the following fields and buttons:

- Username
- Password
- Iniciar (Login)

Esta es la página de Inicio de Sesión, para que ingreses con tu número de usuario y contraseña. Es importante mencionar que si no tienes cuenta primero debes registrarte. Si te registras y no hay ninguna cuenta relacionada con el email proporcionado se te redirigirá a esta ventana y en un mensaje se mostrará tu número de usuario, el cual debes recordar, este es tu "Username". Si el email proporcionado ya está registrado el mensaje dirá "Correo ya registrado"; deberás registrarte con otro correo.

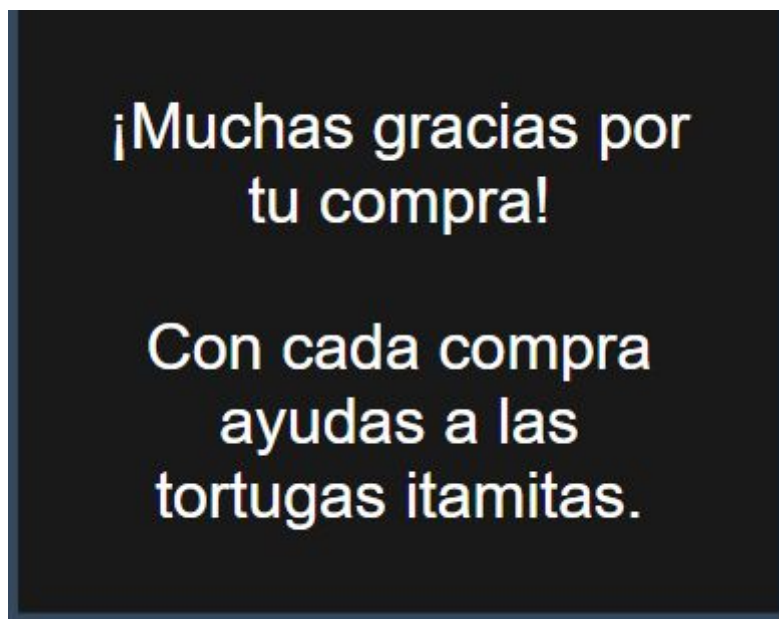
The toolbar contains two buttons:

- Ordenar (Order)
- Log-Out

Esta es la barra de herramientas de la página web una vez que se haya ingresado a una cuenta. Si presionas "Ordenar" te manda a la página donde está el menú para hacer un pedido. Si presionas "Log-Out" regresas a la página de "Inicio" y se cierra tu sesión.



Esta es la página de Ordenar, donde podrás realizar un pedido. Para ello debes seguir tres pasos: 1) elegir la cantidad deseada de cada producto; 2) presionar el botón “Ver Carrito” para verificar tu orden, este paso es IMPORTANTE; 3) una vez que veas que tu orden es la deseada debes presionar el botón “Confirmar Compra” para generar tu pedido.



Tras confirmar un pedido se mostrará esta ventana de agradecimiento. Podrás realizar otro pedido o cerrar tu sesión desde la barra de herramientas.