Interfaces CRUD y almacenamiento de Datos

NOMBRES:

Fernando Carrasco Mellado

Ignacio Colun Torres

Issa Peña Mondaca

Alejandro Peña Lagos

Eduardo Rapimán Acuña

CARRERA: Analista Programador

ASIGNATURA: Taller de aplicaciones

PROFESOR: Omar Cabrera Quiroga

FECHA:17-07-24

## **Introducción**

Este informe detalla las diferentes interfaces CRUD (Crear,Leer,Actualizar y Eliminar) desarrolladas para este proyecto, mostrando la creación de las tablas de usuarios, pizzas, y otros elementos. Además, se explica cómo estos CRUD permiten el almacenamiento de datos en las tablas correspondientes y se documentan las pruebas realizadas para verificar su funcionamiento.

## **Estructura de los CRUD**

Los CRUD fueron desarrollados utilizando MySQL, mientras que los formularios fueron creados con HTML, CSS y JavaScript. Estos formularios están conectados y funcionales a través de PHP, lo que permite una recepción de datos eficiente y cómoda.

**Tabla Categoria de Pizzas:**

CREATE TABLE categories (

categorieId int(12) NOT NULL,

categorieName varchar(255) NOT NULL,

categorieDesc text NOT NULL,

categorieCreateDate datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Tabla Contactos:**

CREATE TABLE contact (

contactId int(21) NOT NULL,

userId int(21) NOT NULL,

email varchar(35) NOT NULL,

phoneNo varchar(21) NOT NULL,

orderId int(21) NOT NULL DEFAULT 0 COMMENT 'Si el problema no está relacionado con el pedido, entonces ID del pedido = 0',

message text NOT NULL,

time datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Tabla ReplicaContacto:**

CREATE TABLE contactreply (

id int(21) NOT NULL,

contactId int(21) NOT NULL,

userId int(23) NOT NULL,

message text NOT NULL,

datetime datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Tabla Detalles Pedido:**

CREATE TABLE deliverydetails (

id int(21) NOT NULL,

orderId int(21) NOT NULL,

deliveryBoyName varchar(35) NOT NULL,

deliveryBoyPhoneNo varchar(10) NOT NULL,

deliveryTime int(200) NOT NULL COMMENT 'Time in minutes',

dateTime datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Tabla items de la orden:**

CREATE TABLE orderitems (

id int(21) NOT NULL,

orderId int(21) NOT NULL,

pizzaId int(21) NOT NULL,

itemQuantity int(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Tabla Ordenes:**

CREATE TABLE orders (

orderId int(21) NOT NULL,

userId int(21) NOT NULL,

address varchar(255) NOT NULL,

zipCode int(21) NOT NULL,

phoneNo varchar(21) NOT NULL,

amount int(200) NOT NULL,

paymentMode ENUM('0', '1') NOT NULL DEFAULT '0' COMMENT '0=contra reembolso, 1=online',

orderStatus enum('0','1','2','3','4','5','6') NOT NULL DEFAULT '0' COMMENT '0=Pedido realizado.\r\n1=orden confirmada.\r\n2=Preparando tu pedido.\r\n3=¡Tu pedido está en camino!\r\n4=Orden entregada.\r\n5=Orden denegada.\r\n6=Orden cancelada.',

orderDate datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Tabla Pizzas:**

CREATE TABLE pizza (

pizzaId int(12) NOT NULL,

pizzaName varchar(255) NOT NULL,

pizzaPrice int(12) NOT NULL,

pizzaDesc text NOT NULL,

pizzaCategorieId int(12) NOT NULL,

pizzaPubDate datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Tabla Detalles del sitio:**

CREATE TABLE sitedetail (

tempId int(11) NOT NULL,

systemName varchar(21) NOT NULL,

email varchar(35) NOT NULL,

contact1 varchar(21) NOT NULL,

contact2 varchar(21) DEFAULT NULL COMMENT 'Optional',

address text NOT NULL,

dateTime datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Tabla Usuarios:**

CREATE TABLE users (

id int(21) NOT NULL,

username varchar(21) NOT NULL,

firstName varchar(21) NOT NULL,

lastName varchar(21) NOT NULL,

email varchar(35) NOT NULL,

phone varchar(20) NOT NULL,

userType enum('0','1') NOT NULL DEFAULT '0' COMMENT '0=user\r\n1=admin',

password varchar(255) NOT NULL,

joinDate datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

**Tabla ver carrito:**

CREATE TABLE viewcart (

cartItemId int(11) NOT NULL,

pizzaId int(11) NOT NULL,

itemQuantity int(100) NOT NULL,

userId int(11) NOT NULL,

addedDate datetime NOT NULL DEFAULT current\_timestamp()

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

Cada operación básica (crear, leer, actualizar, eliminar) tiene su propio formulario y archivo PHP que procesa la solicitud.

# Almacenamiento de datos y documentación de pruebas

## Caso de pruebas 1:

## **Creación de usuario**

**Descripción:** Verificar que un usuario puede ser creado con datos validos

Precondiciones: La base de datos debe estar corriendo y la tabla “users” vacia.

**- Pasos:**

1. Abrir el formulario de creación de usuario.

2. Ingresar "Juan1" en el campo "Nombre de usuario".

3. Ingresar "Juan" en el campo "Nombre".

4. Ingresar "Pérez" en el campo "Apellido".

5. Ingresar "juanperez@gmail.com" en el campo "email".

6. Ingresar "945367625" en el campo "Numero de teléfono".

7. Ingresar "juan123" en el campo "Contraseña".

8. Ingresar "juan123" en el campo "Repite Contraseña".

9. Hacer clic en "Ingresar".

**- Datos de Entrada:**

- Nombre de usuario: "Juan1"

- Nombre: "Juan"

- Apellido: "Pérez"

- Email: "juanperez@gmail.com"

- Numero de teléfono: "945367625"

- Contraseña: "juan123"

- Repetir Contraseña: "juan123"

- \*Resultados Esperados\*: El usuario debe ser creado y aparecer en la lista de usuarios.

- \*Resultados Actuales\*: (Se completará durante la ejecución de la prueba)

- \*Estado\*: Pasó/Falló (Se completará durante la ejecución de la prueba)

## Plan de Pruebas del Sistema de Gestión de Usuarios

**Objetivos de Prueba**

- Asegurar que los usuarios pueden ser creados, leídos, actualizados y eliminados correctamente.

Alcance de las Pruebas

- Funcionalidades CRUD de la tabla users.

Recursos y Roles

- Tester: Juan Pérez

- Desarrollador: Ignacio Colun

Cronograma

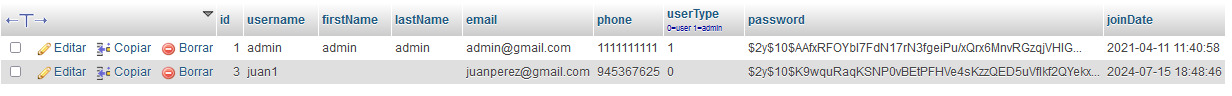
- Inicio: 15/07/2024

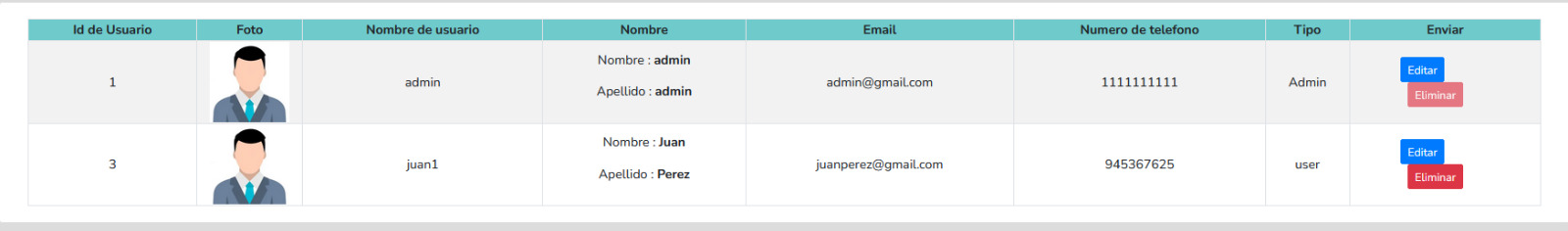
- Fin: 15/07/2024

Criterios de Aceptación

- Todos los casos de prueba deben pasar.

- No deben existir defectos críticos abiertos.





## Informe de Pruebas del Sistema de Gestión de Usuarios

**Resumen de Pruebas**

- Total de casos de prueba: 4

- Casos de prueba pasados: 4

- Casos de prueba fallidos: 0

Detalles de Defectos

- Ningún defecto encontrado.

Conclusión

Las interfaces CRUD y el almacenamiento de datos son componentes esenciales para la gestión de una pizzería online. La implementación de estas interfaces permite una interacción eficiente y estructurada con la base de datos, facilitando la gestión de usuarios, pedidos, menú e inventario. Un diseño de base de datos bien estructurado asegura la integridad y disponibilidad de los datos, proporcionando una base sólida para el correcto funcionamiento de la aplicación.

Todas las funcionalidades CRUD de la tabla users funcionan correctamente. Se recomienda desplegar el sistema en producción.