

Manual de Usuario



Tabla de contenido

Introduccion.....	1
La idea inicial	3
Estrategia.....	¡Error! Marcador no definido.
Bases de Datos	4
BackEnd	17

The Vintage

Introduccion

El e-commerce o comercio electrónico, según la definición de Wikipedia es...

Un sistema de compra y venta de productos y servicios que utiliza Internet como medio principal de intercambio, en otras palabras, se trata de un comercio que gestiona los cobros y pagos a través de medios electrónicos.

Formas de comercio electrónico

El mundo del e-commerce es muy amplio y se pueden encontrar muchos tipos distintos de modelos.

Vamos a hacer 2 clasificaciones:

- Una más global en función de quién vende y quién compra.
- Otra con diferentes «modelos» de e-commerce.

Según el perfil comercial

Cada negocio tiene un tipo de cliente al que se dirige, y en función de ello podemos clasificarlos:

1. B2B (Business-to-Business): empresas cuyos clientes finales son otras empresas u organizaciones. Un ejemplo podría ser una tienda de venta de materiales de construcción que se dirige a interioristas o arquitectos.
2. B2C (Business-to-Consumer): empresas que venden de manera directa a los consumidores finales del producto o

servicio. Es el más habitual y hay miles de ejemplos de tiendas de moda, zapatos, electrónica, etc.

3. C2B (Consumer-to-Business): portales en los que los consumidores publican un producto o servicio y las empresas pujan por ellos. Son los clásicos portales de empleo freelance como Freelancer, Twago, Nubelo o Adtriboo.

4. C2C (Consumer-to-Consumer): empresa que facilita la venta de productos de unos consumidores a otros. El ejemplo más claro sería el de eBay, Wallapop o cualquier otro portal de venta de 2º mano.

Paso1: La idea inicial

El primer paso consistió en escribir las ideas iniciales hasta quedar con una o unas pocas. La idea de e-commerce que analizamos:

- Propuesta de valor: ¿qué valor aporta? ¿Qué necesidad satisface?
- Mercado: ¿es un producto nuevo? ¿Es un producto que ya existe, pero con nuevas funcionalidades?
- Demanda: ¿está ese problema que vas a resolver lo suficiente extendido como para que haya demanda suficiente?
- Dificultades: lista otro retos que surgieron (dificultades técnicas, competencia, amenazas...).

Paso 2: Estrategia

- ¿Cómo nos va a conocer nuestro cliente ideal?
- ¿Por qué nos va a preferir a la competencia?
- ¿Vamos a ser más baratos?
- ¿Qué define a nuestros clientes? ¿Qué características comunes tienen?

Paso 3: Bases de Datos

Se creó la base de datos con la siguiente estructura:

Name	Engine	Version	Row Format
tcliente	InnoDB	10	Dynamic
tinventario	InnoDB	10	Dynamic
torden	InnoDB	10	Dynamic
tperfil	InnoDB	10	Dynamic
tproducto	InnoDB	10	Dynamic
tusuario	InnoDB	10	Dynamic

Cada tabla con la siguiente estructura:

Table: tusuario

Columns:

<u>tidusuario</u>	varchar(18) PK
tusunom	varchar(60)
tusuape	varchar(60)
tusutip	varchar(45)
tusupwd	varchar(45)
tususts	char(1)

Table: tproducto

Columns:

<u>idtproducto</u>	int(11) AI PK
tpronom	varchar(45)
tprocat	varchar(45)
tprodsc	varchar(300)
tprofec	datetime
tprosts	char(1)
tproimg	varchar(300)

Table: tperfil

Columns:

<u>tidperfil</u>	int(11) AI PK
tperusu	varchar(15)

Table: **torden**

Columns:

<u>idtorden</u>	int(11) AI PK
tordcli	int(11)
tordpro	int(11)
tordcant	int(11)
tordamt	int(11)
tordsts	char(1)

Table: **tinventario**

Columns:

<u>tidinventario</u>	int(11) AI PK
tinvcod	int(11)
tinvcan	int(11)
tinvs	char(1)
tinvsiz	varchar(2)
tinvam	int(11)

Table: **tcliente**

Columns:

<u>idcliente</u>	varchar(18) PK
tcliusu	varchar(45)
tclinom	varchar(60)
tcliap1	varchar(60)
tcliem1	varchar(45)
tclite1	int(11)
tclidir	varchar(120)
tclinit	varchar(10)
tclifna	date
tclists	char(1)
tclifec	date
tclipwd	varchar(45)

Se Creó una función para validar el login a nuestra aplicación:

```
DELIMITER $$
CREATE DEFINER='root'@'localhost' FUNCTION "FTN_LOGIN" (usuario varchar(18),pwd varchar(25)) RETURNS char(1) CHARSET utf8
BEGIN
    declare usr varchar(18);
    declare pwd1 varchar(25);
    declare res char;
    set usr = (select tidusuario from tusuario where tidusuario = usuario);
    set pwd1 = (select tusupwd from tusuario where tidusuario = usuario);

    IF exists(select tidusuario from tusuario where tidusuario = usuario) then
        If (usr = usuario)then
            If (pwd1 = pwd)then
                set res = 'I';
            end if;
        end if;
    end if;

    return res;
END$$
DELIMITER ;
```

De esta manera si devuelve “I” esto quiere decir que el usuario y contraseña es correcto, de lo

contrario no nos devuelve nada y esto se interpreta del lado de php para no redirigir a la página de administración.

Se creó también un stored procedure para el ingreso, actualización y eliminación del producto a nuestro Kardex.

```
CREATE DEFINER='root'@'localhost' PROCEDURE 'mant_prod'(  
  IN ACT CHAR(1),  
  IN identprod int,  
  IN nombre varchar(200), /* VACIO SI SE ESTA INSERTANDO*/  
  IN categoria varchar(200),  
  IN descripcion varchar(300),  
  IN imagen varchar(300),  
  OUT MENSAJE VARCHAR(300)  
)  
  
BEGIN  
  DECLARE EXIT HANDLER FOR SQL_EXCEPTION  
  
  BEGIN  
    ROLLBACK;  
    set MENSAJE = 'An error has occurred, operation rollbacked and the stored procedure was terminated';  
  END;  
  /*DECLARE PVID VARCHAR(45);  
  SET PVID = FLOOR(RAND() * 9999999) + 1000000;*/  
  
  START TRANSACTION;  
  If (ACT = 'I' ) then  
    INSERT INTO 'tn'..'tproducto' ('tpronom','tprocat','tprodsc','tprofec','tprosts','tproimg' )  
    VALUES (nombre,categoria,descripcion,now(), 'A',imagen);  
    set MENSAJE = 'PRODUCTO CREADO CORRECTAMENTE';  
  end if;
```

Esto nos permite además de tener un manejo de errores, nos permite también tener integridad del lado de la base de datos para que no se ingrese información incompleta y realizar validaciones de los datos de los productos. Esto nos devuelve mensajes si fue exitoso o no el ingreso de la información.

Paso 4: Back-End

Se utilizó php y JavaScript para este fin y se enlazó con la base de datos antes descrita.

El método de conexión es MSQLI y el envío de la información es por método post

Los archivos utilizados son los siguientes:

conexion.php

Incluye los datos de conexión a nuestra base de datos, esto se realiza de esta manera para garantizar que no puedan leer esta información para acceder a la misma.

```
<?php

$dbhost = "localhost:3307";
$dbuser = "root";
$dbpass = "abc123**";
$dbname = "tn";

$conn = mysqli_connect($dbhost,$dbuser,$dbpass,$dbname);

if(!$conn)
{
    die("No hay conexion:" .mysqli_connect_error());
}
?>
```

funciones.php

Aloja todas las funciones utilizadas dentro de nuestro proyecto

```
<?php
include("conexion.php");

function consultar ($consulta){
    include("conexion.php");
    if (!($datos = mysql_query($com,$consulta))){
        return false; // si fue rechazada la consulta por errores de sintaxis, o ningún registro coincide con lo buscado, devolvemos false
    } else {
        return $datos; // si se obtuvieron datos, los devolvemos al punto que fue llamada la función
    }
}

function tabular ($datos){
    //Abrimos la etiqueta table una sola vez:
    $codigo .= ' <table border="1" cellpadding="3">';

    //Vamos acumulando de a una fila "tr" por vuelta:
    while ($fila = @mysql_fetch_array($datos) ) {

        $codigo .= ' <tr>';

        //vamos acumulando tantos "td" como sea necesario:
        $codigo .= ' <td>' . utf8_encode($fila["trpronon"]) . ' </td>';
    }
}
```

index_producto1.php

esta es la página que tiene el formulario de ingreso de producto a nuestro Kardex.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>e-commerce Team Nocturno</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="">
    <!-- [if lt=8]>
    <http-equiv="X-UA-Compatible" />
    </endif-->
    <!-- bootstrap -->
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
    <link href="bootstrap/css/bootstrap-responsive.min.css" rel="stylesheet">

    <link href="themes/css/bootstrappage.css" rel="stylesheet"/>

    <!-- global styles -->
    <link href="themes/css/flexslider.css" rel="stylesheet"/>
    <link href="themes/css/main.css" rel="stylesheet"/>

    <!-- scripts -->
    <script src="themes/js/jquery-1.7.2.min.js"></script>
    <script src="bootstrap/js/bootstrap.min.js"></script>
    <script src="themes/js/superfish.js"></script>
    <script src="themes/js/jquery.scrollTo.js"></script>
    <script src="themes/js/3prod.js"></script>
    <!-- [if lt=8] -->
    <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
    <script src="js/respond.min.js"></script>
    </endif-->
</head>
```

index_prod.php

Aquí tenemos toda la programación para el ingreso de los datos del formulario de ingreso de producto a nuestro Kardex y toda la lógica para que no se ingresen ningún otro archivo que no sea imagen.

```
<?php
include("conexion.php");

$Prodcod = $_POST["Prodcod"];
$ProdNom = $_POST["ProdNom"];
$categoria = $_POST["categoria"];
$imagen = $_POST["imagen"];
$descripcion = $_POST["descripcion"];
$fecha = date("Y/m/d");
$imagen2 =
if(isset($_POST["btnguardar"]))
{
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO tproducto(`tpronom`,`tprocat`,`tprodsc`,
                                `tprofec`,`tprosts`,`tproimg`)
VALUES
('$Prodcod','$categoria','$descripcion','$fecha','A','$imagen')";

if ($conn->query($sql) === TRUE) {
    echo "Dato Ingresado correctamente";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
}
```

login_registrar.php

En este archivo encontramos las validaciones del login de usuarios administrativos, vital para crear el login hacia la página de administración

```
<?php

include("conexion.php");

$nombre = $_POST["usuario"];
$pass   = $_POST["pass"];

//Login
if(isset($_POST["btningresar"]))
{
    $consulta = "select ftn_login('$nombre','$pass')";
    $query = mysqli_query($conn,$consulta);
    if($fila = mysqli_fetch_array($query))
    {
        echo "<script> alert('Bienvenido $nombre'); window.location='index_producto1.php' </script>";
    }else
    {
        echo "<script> alert('Usuario no existe'); window.location='index.html' </script>";
    }
}

?>
```

Carrito.js

Esta clase consiste en guardar los datos por cada producto que se escoge cada método tiene una función a tal punto de modelar los el articulo.

```
class Carrito {  
  //Añadir producto al carrito  
  comprarProducto(e){  
    e.preventDefault();  
    //Delegado para agregar al carrito  
    if(e.target.classList.contains('agregar-carrito')){  
      const producto = e.target.parentElement.parentElement;  
      //Enviamos el producto seleccionado para tomar sus datos  
      this.leerDatosProducto(producto);  
    }  
  }  
  
  //Leer datos del producto  
  leerDatosProducto(producto){  
    const infoProducto = {  
      imagen : producto.querySelector('img').src,  
      titulo: producto.querySelector('h4').textContent,  
      precio: producto.querySelector('.precio span').textContent,  
      id: producto.querySelector('a').getAttribute('data-id'),  
      cantidad: 1  
    }  
    let productosLS;  
    productosLS = this.obtenerProductosLocalStorage();  
    productosLS.forEach(function (productoLS){  
      if(productoLS.id === infoProducto.id){  
        productosLS = productoLS.id;  
      }  
    });  
  
    if(productosLS === infoProducto.id){
```


Compra.js

Este js carga los Eventos del mouse al seleccionar el producto este hace un llamado a la clase carrito.js el cual guarda los datos del producto

```
const compra = new Carrito();
const listaCompra = document.querySelector("#lista-compra tbody");
const carrito = document.getElementById('carrito');
const procesarCompraBtn = document.getElementById('procesar-compra');
const cliente = document.getElementById('cliente');
const correo = document.getElementById('correo');

cargarEventos();

function cargarEventos() {
  document.addEventListener('DOMContentLoaded', compra.leerLocalStorageCompra());

  //Eliminar producto del carrito
  carrito.addEventListener('click', (e) => { compra.eliminarProducto(e) });

  compra.calcularTotal();

  //Cuando se selecciona procesar Compra
  procesarCompraBtn.addEventListener('click', procesarCompra);

  carrito.addEventListener('change', (e) => { compra.obtenerEvento(e) });
  carrito.addEventListener('keyup', (e) => { compra.obtenerEvento(e) });
}

function procesarCompra() {
  // alertar de exito;
  if (compra.obtenerProductosLocalStorage().length >= 0) {
    Swal.fire({

```

Paso 5: Versionamiento o repositorio.

Se ha utilizado el sistema de GITHUB el cual explicamos brevemente a continuación:

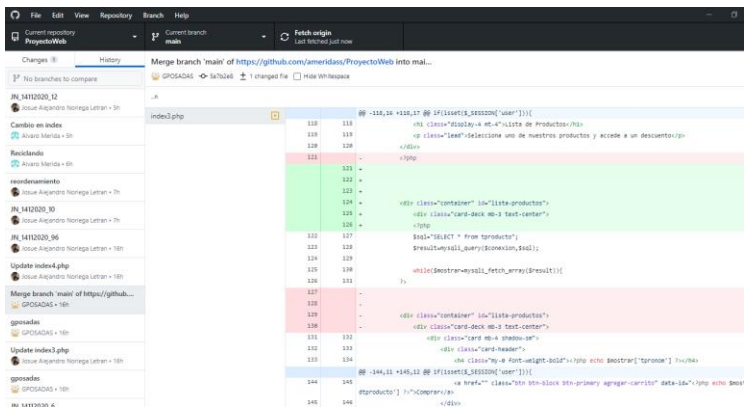
“GitHub es un sistema de gestión de proyectos y control de versiones de código, así como una plataforma de red social diseñada para desarrolladores. ¿Pero para qué se usa GitHub? Bueno, en general, permite trabajar en colaboración con otras personas de todo el mundo, planificar proyectos y realizar un seguimiento del trabajo.”

Esto nos permite que varios desarrolladores trabajen sobre el mismo proyecto y tengan los cambios de los demás para sus pruebas.

Esto lo implementamos para poder realizar este desarrollo cada uno realizando la carga de la parte correspondiente.

El servicio se cargó en una instalación de XAMMP, lo cual nos permitió tener el manejo de la base de datos localmente al igual que el alojamiento de nuestra pagina ya que se

utilizaba HTML y PHP y este servidor local lo carga automáticamente, lo único que se configuró fue el puerto de conexión a Mysql.



Autores

Alvaro Sosa

5646

Josue Noriega

1597874590406

Geizer Posadas

202002068