**ORGANIZACIÓN GENERAL**

1.-Estructurar base de datos(tablas y relaciones entre ellas).

2-Cuántas páginas CONTEMPLADAS (4)

3.-Pseudocódigo

4-Empezar a realizar

* Formularios
* Headers y action forms(cambio entre páginas)
* Sesiones (iniciar y cerrar sesión de todos)
* Cookies y amonestaciones (pedido fuera de tiempo y cancelación del mismo)
* Insertar valores (hacer pedidos, agregar productos)
* Consultas (Ver qué pedidos están vigentes, qué productos disponibles, consulta de estado en pedidos)
* Alterar valores (Cambiar info de usuario, modificar productos, cambiar estado pedido)
* Eliminar (Regular cant usuarios) ((‘Borrar registros’))

5-Seguridad.

6.-Documentación. /

7.-Estilos./

8.-Extras. /

ITINERARIO:

21- Redactar pseudocodigo

22-Terminamos de redactar y diseñar interfaz del mr administrador y supervisor

23-Formularios y headers y action forms/Empezar con sesiones

24-Regex

25-Sesiones y cookies

26-Toda consulta, inserción, modificación y eliminación.

27- Documentación y continuar con consultas

28 Juntar y estilos documentación

29 Juntar y Terminado documentación

30 Entrega y detalles extra

PÀGINAS:

Registro.php

Ingreso.php

CoyoRappi.html

Ayuda.html

(PEDIDOS.PHP)

**(1) Ingreso (Inicio de sesión)**

Ingreso Tipo de Usuario: (<select name=tipoUsuario>)

* Alumno.
* Profesor o funcionario.
* Trabajador.
* Administrador.
* Supervisor.<option value>

Ingresar Usuario

* Alumno: Insertar el número de cuenta nc
* Profesor o funcionario: Insertar el RFC
* Trabajador: Insertar el número de trabajador. nt
* Administrador:Usuario y contraseña predefinidas
* Supervisor:Usuario y contraseña predefinidas

Ingresar Contraseña (hasheada)

Hacer conexión con base de datos y verificar que usuario exista y que la contraseña coincida.

En caso de no coincidir o no existir se alerte al usuario que esta no existe.

Botón de Inicio <input type= ‘submit’ method=”POST” action=”Pedido.php”> para redireccionar el pedido.

Botón de Registro para redireccionar a Registro.

**(2) Registro:**

**(3) Function verificar() ($campo, $tabla,$valor){**

**$a=0;**

**If(el valor en el campo tal en tabla tal existe){**

**$a=1**

**}**

**return=$a**

**}**

(4) Form

Campo de tipo de cliente (Alumno(1), Profe/Estético(3), Trabajador(2))

Te redirecciona a tu página de registro dependiendo el valor de cliente(modelo rodkollega)

Si vale 1 (alumno){

Nombre text

Grupo number

No.cuenta

(Contraseña)(Pendiente)

Hacer conexión con la base de datos y verificar()

Si $a=0 {

INSERT en Alumnos los valores

Header a pedidos

}

Else{

echo”Ese usuario ya existe, ingrese sesión o ponga uno válido”

Link para ingreso

}

}

Elseif vale 3 (Profe/Funcionario){

Nombre

Colegio

RFC

Contraseña

Hacer conexión con la base de datos y verificar()

Si $a=0 {

INSERT en Profe los valores

Header a pedidos

}

Else{

echo”Ese usuario ya existe, ingrese sesión o ponga uno válido”

Link para ingreso

}

}

Else if vale 2 (trabajador){

Número de trabajador

Nombre

Contraseña.

Hacer conexión con la base de datos y verificar()

Si $a=0 {

INSERT en Profe los valores

Header a pedidos

}

Else{

echo”Ese usuario ya existe, ingrese sesión o ponga uno válido”

Link para ingreso

}

}

**(5) Pedidos**

(6) Verificar tipo de usuario{

Si es administrador header=(Administrador.php)

Si es supervisor header=(Supervisor.php)

}

Else

{

**Si existe la variable de sesión de calcular y no vacía entonces imprima total y**

**Echo”alimento y cantidad”**

* Lugar de entrega

Fecha(hidden)---- Recordar en la instalación que se modifique la configuración en .init

Hora(hidden)

estado(hidden) default ‘activo’

echo”Session total”

Input submit value pedir form action=’pedir.php’

Pedir{

INSERT VALUES TAL EN TABLA PEDIDOS

Y header cerrar.php

}

Cerrar.php{

Asignar vacío a solamente la variable calcular y la variable de disponibilidad y header a pedidos.

}

**}**

**else{**

**Si existe la variable disponibilidad y no vacía entonces{**

* calcular(){
* Crea una variable de sesión total y calcula el total después con un header te manda a pedidos
* }

}

**else{**

**Si no existe entonces desplegar menú de pedido normal**

Campos:

Select from alimento where disponibilidad != 0, para cada registro crear opción con nombre(

* Alimento(/paquete) )
* Cantidad
* Input submit verificar() FORM action=(verificar.php (function)) method= (POST)
* Verificar{
* Si disponibilidad>= cantidad en tabla alimentos en id\_alimentos, entonces sesion start crea variable de sesion disponible, envia la informacion y haz header a pedidos

UPDATE Alimentos SET Disponibilidad=”la que este”-$cantidad WHERE Disponibilidad

* }

}

}

● n.

● El supervisor de los pedidos será el encargado de verificar las órdenes, de hacer

entrega de los pedidos y ya sea de finalizar la entrega con éxito al liquidarse la

orden, o de sancionar al usuario por retraso, falta o no liquidar la orden.

El administrador podrá agregar, modificar o eliminar productos de la cafetería, así

como regular la cantidad de usuarios y la modificación de su información.

Interfaz del admin

(Administrador.php)

Agregar, eliminar y modificar elementos de la tabla alimentos, alumno, profe/funcionario, trabajador, así como restringir permisos de los usuarios.

**Agregar productos**

Contraseña del Admin

Nombre, Disponibilidad, Costo, Contraseña

Verificar que contraseña coincida con contraseña mysql

Verificar en mysql si no existe el nombre del alimento

(INSERT INTO Alimentos (Nombre, Disponibilidad,Costo) VALUES (“*Nombredelproducto*”, int(4), float(5))

**Eliminar productos**

Contraseña del Admin

DELETE FROM Alimentos

WHERE Nombre (*Nombre ya existente*)

LIMIT 1

**Modificar productos**

Contraseña del Admin

Campo a modificar

Nuevo valor

Update Alimentos Campo a modificar= “Nuevo valor“ WHERE Campo de referencia ya existente= “valor ya existente”

**Modificar la info de los Usuarios**

Contraseña del Admin

Tipo de usuario

Si es cliente, tipo de cliente

Usuario a modificar

Campo a modificar

Nuevo valor

Si tipo de usuario== cliente y tipo cliente ta vacío entonces echo Bro especifica el tipo de cliente

ELSE

If cliente

UPDATE TABLE tipo de cliente where usuario == $usuario, cambiar campo por nuevo valor

Else update table tipo de usuario where usuario = $usuario, cambiar campo por nuevo valor.

**Eliminar usuarios**

Contraseña del Admin

Tipo de usuario

Si es cliente, tipo de cliente

Usuario a modificar

Campo a modificar

Nuevo valor

Si tipo de usuario== cliente y tipo cliente ta vacío entonces echo Bro especifica el tipo de cliente

ELSE

If cliente

DELETE FROM tipo de cliente

WHERE usuario == $usuario

LIMIT 1

**Agregar adms y supervisores**

Contraseña del Admin

Nuevo valor

Contraseña del nuevo valor

INSERT INTO Administrador (“Usuario”, “Contraseña”) VALUES (“Nuevo valor” , “Contraseña del nuevo valor”

INSERT INTO Supervisor (“Usuario”, “Contraseña”) VALUES (“Nuevo valor” , “Contraseña del nuevo valor”

Nota: Se creará un usuario en la base de datos llamado Admin al que se le otorgarán todos los permisos a excepción de eliminar la base de datos, en los parámetros de instalación se enviará la contraseña cifrada.

INTERFAZ DEL SUPERVISOR(Supervisor.php)

**Modificar el estado de las órdenes**

??????????????????????????????????????????????????????????????????????????????????????????????????????????¡=??

UPDATE Pedidos SET $estado WHERE estado

**Verificar las órdenes**

<input type=”submit” name=”Consultar” action=”mostrarTabla”>

mostrarTabla(){

SELECT fecha,hora pedido, hora entrega, estado, cantidad, total FROM Pedidos;

}

**Sancionar al usuario**

Select tipo de usuario

Ingresar Usuario

Select estadoPedido <activo>, <entregado>

Si el tipoUsuario es Alumno

Verificar que el usuario exista en la tabla Alumnos

UPDATE Alumno SET $estadoPedido WHERE estado(default)

Else

Echo “Verifica que Usuario sea correcto”;

header=>Supervisor.php

Si el tipoUsuario es Trabajador

Verificar que el usuario exista en la tabla trabajador

UPDATE Trabajador SET $estadoPedido WHERE estado(default)

Else

Echo “Verifica que Usuario sea correcto”;

header=>Supervisor.php

Si el tipoUsuario es Profesor || Funcionario

Verificar que el usuario exista en la tabla profe-funcionario

UPDATE ProfeFuncionario SET $estadoPedido WHERE estado(default

Else

Echo “Verifica que Usuario sea correcto”;

header=>Supervisor.php

Cosas que se podrían mejorar ;-; : (

Alerta cuando se va a eliminar datos.

Automatizar las sanciones(no se puede hacer una cookie con esto?, tipo se crea el pedido y si no se ha borrado para la hora que tocaba se crea una cookie para el usuario que lo bloquea y solita se borra con el time?)