




CONTROL DE VERSIONES					
Versión	Hecha por	Revisado por	Aprobada por	Fecha	Motivo
1.0					

Plan de Gestión de calidad de software

Nombre del proyecto	Integrantes
Deployment Programmers 	<ul style="list-style-type: none"> • Yecenia González Moreno • Jesús Chávez Jesús David Chávez • Juan Pablo Merchán Barrera • Karina Bocanegra Cupitra

1.- Definición de requisitos. Alcance del proyecto.

Desarrollar un aplicativo Web que tenga como objetivo lograr que nuestros clientes puedan realizar la migración del proceso tradicional de manejo de información a la era digital. Esto permitirá llevar control en tiempo real de la información y el cronograma de su actividad económica logrando modificar, agregar o eliminar datos.

Éste se desarrollará mediante lenguaje de programación Web PHP y los datos ingresados por medio de los formularios establecidos que será almacenado en BD Mysql.

Contará con opción login por medio de módulo de ingreso; Se requiere el uso de usuario (email) y contraseña.

Internamente contará con módulos de procesos de acuerdo a la necesidad del cliente, ya sea desde agenda hasta la información de sus proveedores.

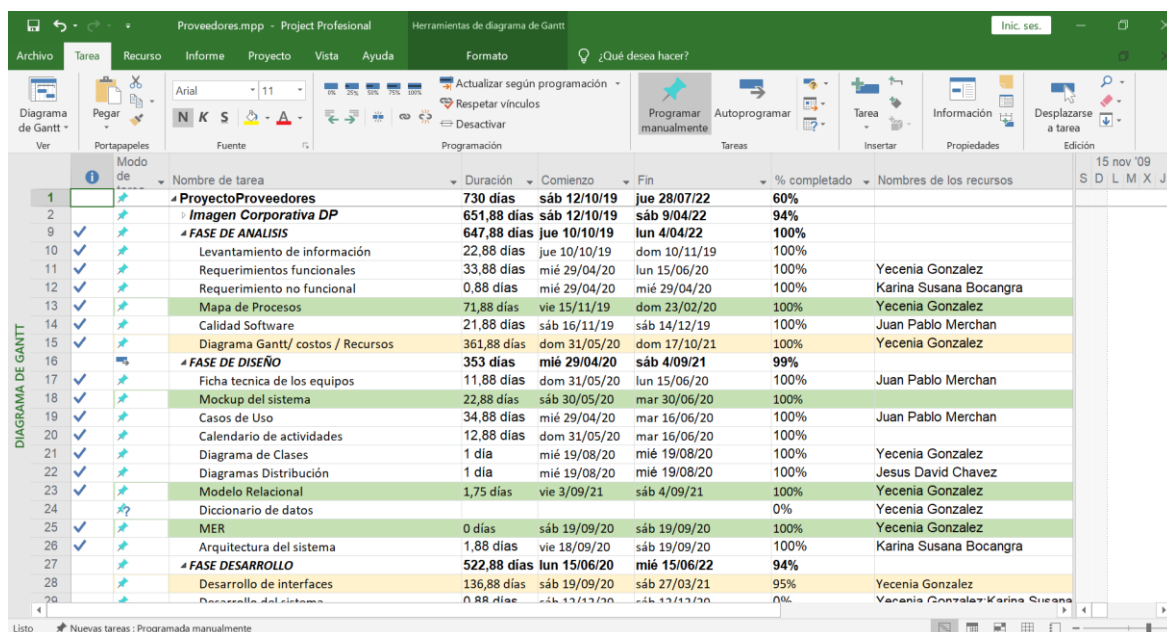
Nota: Los requerimientos en el Formato IEEE.



IEEE.docx

2.- Dividir los desarrollos en fases o entregables.

Se realiza un cronograma por fases de las tareas en la herramienta Project el cual es llamado diagrama de Gantt, este me permite asignar un responsable de la actividad, una fecha de inicio, una fecha final y mediante una grafica me permite visualizar el porcentaje de la actividad.



3.- Elección de un IDE que se adapte a tus necesidades.

Durante el desarrollo del aplicativo web, se utiliza el editor IDE Visual Studio Code, el cual es compatible con múltiples lenguajes de programación, entre ellos desarrollo web, PHP, HTML, CSS, JavaScript, fácil de manejar.

Para las pruebas de automatización se utiliza IntelliJ IDEA java con Selenium.

4.- Estandarizar las reglas del desarrollo.

Una de las reglas de nuestro código es al momento definir una variable nunca con se coloca un carácter especial, el nombre de variable o funciones son autodescriptivas.

Las variables todas están Camel Case.

No hay código comentado.

No hay anotaciones para describir una función.

Nosotros tenemos separados las vistas, controlador, modelo, nunca se visualiza código de backend en una vista.

Para hacer modificaciones generales del headear y footer, se encuentra separado para evitar repetir código.

El llamado de la vista se hace a través del action de la siguiente forma:

?c=nombreControlador ->para llamar una función que solo tenga index

?c=nombreControlador&a=nombreFuncion ->para llamar cualquier función dentro del controlador.

Codigo Modelo / Vista / Controlador

```
<?php
require_once 'Model/UserModel.php';
class UserController
{
    private $modelo;
    public function __construct()
    {
        $this->modelo = new UserModel();
    }
    public function crearUser(){

        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            $usuario = new UserModel();
            $usuario->setRol('R-N/A');
            $usuario->setTipoDoc($_POST['tipoDoc']);
            $usuario->setHorario('h-N/A');
            $usuario->setStatus('Pen');
            $usuario->setNumdocumento($_POST['documento']);
            $usuario->setNombre($_POST['nombre']);
            $usuario->setApellido($_POST['apellido']);
        }
    }
}
```

```

        $usuario->setEmail($_POST['email']);
        $usuario->setTelefono($_POST['telefono']);
        $usuario->setPass(/* password_hash
*/($_POST['password']/*,PASSWORD_DEFAULT */));
        $usuario->setQuestion1($_POST['question1']);
        $usuario->setQuestion2($_POST['question2']);
        $usuario->setQuestion3($_POST['question3']);
        $usuario->setTyc(1);
        $usuario->setTd(1);

        echo $this->modelo->registrar($usuario);
        header('Location: ?c=LandingPageWeb&a=register');
    }
}

public function crearUserAdmin(){
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {

        $usuario = new UserModel();
        $usuario->setRol($_POST['tipoRol']);
        $usuario->setTipoDoc($_POST['tipoDoc']);
        $usuario->setHorario($_POST['horario']);
        $usuario->setStatus($_POST['status']);
        $usuario->setNumdocumento($_POST['documento']);
        $usuario->setNombre($_POST['nombre']);
        $usuario->setApellido($_POST['apellido']);
        $usuario->setEmail($_POST['email']);
        $usuario->setTelefono($_POST['telefono']);
        $usuario->setPass(password_hash($_POST['password'],
PASSWORD_DEFAULT));
        $usuario->setPass2(sha1($_POST['password2']));
        $usuario->setQuestion1($_POST['question1']);
        $usuario->setQuestion2($_POST['question2']);
        $usuario->setQuestion3($_POST['question3']);
        $usuario->setTyc($_POST['tyc']);
        $usuario->setTd($_POST['td']);

        if (($usuario->setPass(password_hash($_POST['password'],
PASSWORD_DEFAULT)) === $usuario->setPass2(sha1($_POST['password2'])) &&
($usuario->setPass(password_hash($_POST['password'], PASSWORD_DEFAULT)) !=
"") && ($usuario->setPass2(sha1($_POST['password2'])) != "")) {
            echo "La informacion de la contraseña coincide";

            echo $this->modelo->registrar($usuario);
            header('Location: ?c=ModulsAdminController&a=users');
        } else {

```

```

        echo "<script> alert('La informacion de la contraseña no
coincide o es vacia') </script>";
        header('Location:?c=ModulsAdminController&a=users');
    }
}

if ($_SERVER['REQUEST_METHOD'] == 'GET') {

    require_once 'view/roles/admin/menuLateralProveedores.view.php';
    require_once 'view/modules/adminUser/usuarios.view.php';
    require_once 'view/roles/admin/footerAdmin.php';
}
}
public function listar(){

    $users= $this->modelo->consultarUserAdmin();

    require_once 'view/roles/admin/headAdmin.php';
    require_once 'view/roles/admin/menuLateralProveedores.view.php';
    require_once 'view/modules/adminUser/searchUsers.view.php';
    require_once 'view/roles/admin/footerAdmin.php';
}
public function updateUser(){

    if(($_SERVER['REQUEST_METHOD'])=='GET'){
        $user= $this->modelo->getById($_GET['idUsuario']);
        $roles= ['No Aplica','Administrador','Auxiliar','Asesor'];
        $estados=['Activo','Inactivo','Pendiente','Cancelado','Reagendad
o'];

        $horarios=['No Aplica','Mañana','Tarde','Rotativo'];
        $tipoDoc=['No Asignado','Cedula de Ciudadania','Cedula de
Ciudadania','Registro Unico Tributario','Numero de Identificacion Tributaria
','Cedula de Extranjeria'];
        require_once 'view/roles/admin/headAdmin.php';
        require_once 'view/roles/admin/menuLateralProveedores.view.php';
        require_once 'view/modules/adminUser/userUpdate.view.php';
        require_once 'view/roles/admin/footerAdmin.php';
    }
    if(($_SERVER['REQUEST_METHOD'])=='POST'){
        $userData = new UserModel();
        $userData->setIdUsuario($_POST['idUsuario']);
        $userData->setTipoDoc($_POST['tipoDoc']);
        $userData->setNumdocumento($_POST['numDocumento']);
        $userData->setEmail($_POST['email']);
    }
}

```

```

        $userData->setHorario($_POST['horario']);
        $userData->setStatus($_POST['estado']);
        $userData->setNombre($_POST['nombre']);
        $userData->setApellido($_POST['apellido']);
        $userData->setRol($_POST['tipoRol']);
        $userData->setTelefono($_POST['telefono']);
        $userData->setPass(password_hash($_POST['password'],
PASSWORD_DEFAULT));
        $userData->setQuestion1($_POST['question1']);
        $userData->setQuestion2($_POST['question2']);
        $userData->setQuestion3($_POST['question3']);
        $userData->setTyc(1);
        $userData->setTd(1);

        if (($userData->setPass(password_hash($_POST['password'],
PASSWORD_DEFAULT)) === $userData->setPass2(sha1($_POST['password2'])) &&
($userData->setPass(password_hash($_POST['password'], PASSWORD_DEFAULT)) !=
"") && ($userData->setPass2(password_hash($_POST['password'],
PASSWORD_DEFAULT)) != ""))) {
            /* var_dump($userData);
            print_r($userData);
            */
            $this->modelo->actualizarUser($userData);
            header('Location:?c=UserController&a=listar');

        } else {

            echo "<script> alert('La informacion de la contraseña no
coincide o es vacia') </script>";
            header('Location:?c=ModulsAdminController&a=users');
        }

    }

}

public function deleteUser(){

    if(($_SERVER['REQUEST_METHOD']=='GET'){
        $user= $this->modelo->getById($_GET['idUsuario']);
        require_once 'view/roles/admin/headAdmin.php';
        require_once 'view/roles/admin/menuLateralProvedores.view.php';
        require_once 'view/modules/adminUser/deleteUser.view.php';
        require_once 'view/roles/admin/footerAdmin.php';
    }
    $this->modelo->deleteUser($_GET['idUsuario']);

```

```
        header('Location:?c=UserController&a=listar');
    }
}
```

```
<?php

class UserModel
{

#selection
    private $idUserario;
    private $nombre;
    private $apellido;
    private $tipoDoc ;
    private $numDocumento;
    private $email;
    private $password;
    private $password2;
    private $telefono;
    private $question1;
    private $question2;
    private $question3;
    private $status;
    private $rol;
    private $tyc;
    private $td ;
    private $horario;

#end
    private $conn;
    public function __construct()
    {
        $a = func_get_args();
        $i = func_num_args();
        if (method_exists($this, $f= '__construct' . $i)) {
            call_user_func_array(array($this, $f), $a);
        }
    }
    function __construct0()
    {
        try {
```

```
        $this->conn = ConexionDB::conexion();
    } catch (Exception $e) {
        die($e->getMessage());
    }
}

function __construct2($email,$password)
{
    $this->email = $email;
    $this->password = $password;
}

public function __construct10($idUserario, $rol, $tipoDoc, $horario,
$status, $numDocumento, $nombre, $apellido, $email, $telefono)
{
    $this->idUserario = $idUserario;
    $this->rol = $rol;
    $this->tipoDoc = $tipoDoc;
    $this->horario = $horario;
    $this->status = $status;
    $this->numDocumento = $numDocumento;
    $this->nombre = $nombre;
    $this->apellido = $apellido;
    $this->email = $email;
    $this->telefono = $telefono;
}

public function __construct16($idUserario, $rol, $tipoDoc, $horario,
$status, $numDocumento, $nombre, $apellido, $email,
$telefono,$password,$question1,$question2,$question3,$tyc,$td)
{
    $this->idUserario = $idUserario;
    $this->rol = $rol;
    $this->tipoDoc = $tipoDoc;
    $this->horario = $horario;
    $this->status = $status;
    $this->numDocumento = $numDocumento;
    $this->nombre = $nombre;
    $this->apellido = $apellido;
    $this->email = $email;
    $this->telefono = $telefono;
    $this->password = $password;
    $this->question1 = $question1;
    $this->question2 = $question2;
    $this->question3 = $question3;
    $this->tyd = $tyc;
    $this->td = $td;
```



```
}  
#section  
public function getIdUsuario()  
{  
    return $this->idUsuario;  
}  
public function setIdUsuario($idUsuario)  
{  
    return $this->idUsuario = $idUsuario;  
}  
public function getRol()  
{  
    return $this->rol;  
}  
public function setRol($rol)  
{  
    return $this->rol = $rol;  
}  
public function getTipoDoc()  
{  
    return $this->tipoDoc;  
}  
public function setTipoDoc($tipoDoc)  
{  
    return $this->tipoDoc = $tipoDoc;  
}  
public function getHorario()  
{  
    return $this->horario;  
}  
public function setHorario($horario)  
{  
    return $this->horario = $horario;  
}  
public function getStatus()  
{  
    return $this->status;  
}  
public function setStatus($status)  
{  
    return $this->status = $status;  
}  
public function getNumdocumento()  
{  
    return $this->numDocumento;  
}
```

```
}  
public function setNumdocumento($numDocumento)  
{  
    return $this->numDocumento = $numDocumento;  
}  
public function getNombre()  
{  
    return $this->nombre;  
}  
public function setNombre($nombre)  
{  
    return $this->nombre = $nombre;  
}  
public function getApellido()  
{  
    return $this->apellido;  
}  
public function setApellido($apellido)  
{  
    return $this->apellido = $apellido;  
}  
public function getEmail()  
{  
    return $this->email;  
}  
public function setEmail($email)  
{  
    return $this->email = $email;  
}  
public function getTelefono()  
{  
    return $this->telefono;  
}  
public function setTelefono($telefono)  
{  
    return $this->telefono = $telefono;  
}  
public function getPass()  
{  
    return $this->password;  
}  
public function setPass($password)  
{  
    return $this->password = $password;  
}
```

```
public function getPass2()
{
    return $this->password2;
}
public function setPass2($password2)
{
    return $this->password2 = $password2;
}
public function getQuestion1()
{
    return $this->question1;
}
public function setQuestion1($question1)
{
    return $this->question1 = $question1;
}
public function getQuestion2()
{
    return $this->question2;
}
public function setQuestion2($question2)
{
    return $this->question2 = $question2;
}
public function getQuestion3()
{
    return $this->question3;
}
public function setQuestion3($question3)
{
    return $this->question3 = $question3;
}
public function getTyc()
{
    return $this->tyc;
}
public function setTyc($tyc)
{
    return $this->tyc = $tyc;
}
public function getTd()
{
    return $this->td;
}
```

```
public function setTd($td)
{
    return $this->td = $td;
}
#endsection

public function registrar($usuario)
{
    try {

        $userSen = ' INSERT INTO usuarios(`id_usuario`,
`cod_rol_usuario`, `cod_tipo_documento_usuario`, `cod_horario_user`,
`cod_estado_usuario`, `numero_documento`, `nombre`, `apellido`, `email`,
`telefono`, `contrasenia`, `question1`, `question2`, `question3`,
`termino_condiciones`,
`tratamiento_datos`)values(null,:cod_rol_usuario,:cod_tipo_documento_usuario
,:cod_horario_user,:cod_estado_usuario,:numero_documento,:nombre,:apellido,:
email,:telefono,:contrasenia,:question1,:question2,:question3,:termino_condi
ciones,:tratamiento_datos)';

        $dbh = $this->conn->prepare($userSen);

        $dbh->bindValue('cod_rol_usuario', $usuario->getRol());
        $dbh->bindValue('cod_tipo_documento_usuario', $usuario-
>getTipoDoc());
        $dbh->bindValue('cod_horario_user', $usuario->getHorario());
        $dbh->bindValue('cod_estado_usuario', $usuario->getStatus());
        $dbh->bindValue('numero_documento', $usuario-
>getNumdocumento());
        $dbh->bindValue('nombre', $usuario->getNombre());
        $dbh->bindValue('apellido', $usuario->getApellido());
        $dbh->bindValue('email', $usuario->getEmail());
        $dbh->bindValue('telefono', $usuario->getTelefono());
        $dbh->bindValue('contrasenia', $usuario->getPass());
        $dbh->bindValue('question1', $usuario->getQuestion1());
        $dbh->bindValue('question2', $usuario->getQuestion2());
        $dbh->bindValue('question3', $usuario->getQuestion3());
        $dbh->bindValue('termino_condiciones', $usuario->getTyc());
        $dbh->bindValue('tratamiento_datos', $usuario->getTd());
        $dbh->execute();

    } catch (Exception $e) {
        die($e->getMessage());
    }
}
```

```
public function consultarUserAdmin()
{
    try {
        $userList = [];

        $bdhUser = $this->conn->query('SELECT * from usuarios');
        foreach($bdhUser->fetchAll() as $user) {
            $userList[] = new UserModel(
                $user['id_usuario'],
                $user['cod_rol_usuario'],
                $user['cod_tipo_documento_usuario'],
                $user['cod_horario_user'],
                $user['cod_estado_usuario'],
                $user['numero_documento'],
                $user['nombre'],
                $user['apellido'],
                $user['email'],
                $user['telefono'],
                $user['contrasenia'],
                $user['question1'],
                $user['question2'],
                $user['question3'],
                $user['termino_condiciones'],
                $user['tratamiento_datos']
            );
        }

        return $userList;
    } catch (Exception $e) {
        die($e->getMessage());
    }
}

public function getByid($idUserario){
    try {

        $sql='SELECT * FROM usuarios WHERE id_usuario=:idUserario';
        $dbh= $this->conn->prepare($sql);
        $dbh->bindValue('idUserario',$idUserario);
        $dbh->execute();
        $searchDB= $dbh->fetch();
        $user= new UserModel(
            $searchDB['id_usuario'],
            $searchDB['cod_rol_usuario'],
            $searchDB['cod_tipo_documento_usuario'],
```

```
        $searchDB['cod_horario_user'],
        $searchDB['cod_estado_usuario'],
        $searchDB['numero_documento'],
        $searchDB['nombre'],
        $searchDB['apellido'],
        $searchDB['email'],
        $searchDB['telefono'],
        $searchDB['contrasenia'],
        $searchDB['question1'],
        $searchDB['question2'],
        $searchDB['question3'],
        $searchDB['termino_condiciones'],
        $searchDB['tratamiento_datos']

    );
    return $user;

} catch (Exception $e) {
    die($e->getMessage());
}

}

public function actualizarUser($userData){

    try {

        $sqlUpdate=$this->conn->prepare("UPDATE `usuarios` SET
`cod_rol_usuario`=:rol,`cod_tipo_documento_usuario`=:tipoDoc,`cod_horario_us
er`=:horario,`cod_estado_usuario`=:status,`numero_documento`=:numDocumento,`
nombre`=:nombre,`apellido`=:apellido,`email`=:email,`telefono`=:telefono,`co
ntrasenia`=:password,`question1`=:question1,`question2`=:question2,`question
3`=:question3,`termino_condiciones`=:tyc,`tratamiento_datos`=:td");

        $sqlUpdate->execute(array(

            ':rol'=> $userData->getRol(),
            ':tipoDoc'=> $userData->getTipoDoc(),
            ':horario'=> $userData->getHorario(),
            ':status'=> $userData->getStatus(),
            ':numDocumento'=> $userData->getNumdocumento(),
            ':nombre'=> $userData->getNombre(),
            ':apellido'=> $userData->getApellido(),
            ':email'=> $userData->getEmail(),
            ':telefono'=> $userData->getTelefono(),
            ':password'=> $userData->getPass(),
```

```

        ':question1'=> $userData->getQuestion1(),
        ':question2'=> $userData->getQuestion2(),
        ':question3'=> $userData->getQuestion3(),
        ':tyc'=> $userData->getTyc(),
        ':td'=> $userData->getTd(),
    ));

    } catch (Exception $e) {
        die($e->getMessage());
    };

}

public function deleteUser($idUserario){
    try {
        $sql= "DELETE FROM `usuarios` WHERE id_usuario=:idUserario";
        $dbh= $this->conn->prepare($sql);
        $dbh->bindValue('idUserario',$idUserario);
        /* $dbh->bindValue('numero_documento',$numDocumento); */
        $dbh->execute();

    } catch (Exception $e) {
        die($e->getMessage());
    };
}

public function initSesion($usuario){
    try {

        $sql = $this->conn->query(" SELECT
id_usuario,cod_rol_usuario,cod_horario_user,cod_estado_usuario,nombre,apelli
do,email,contrasenia,question1,question2,question3 FROM usuarios WHERE
email=

        :email and contrasenia=:password and cod_estado_usuario='Act'");

        $sql->bindValue('email', $usuario->getEmail());
        $sql->bindValue('contrasenia', $usuario->getPass());
        $sql->execute();
        $searchDB=$sql->fetch();

        if ($searchDB) {
            $user = new UserModel(
                $searchDB['id_usuario'],
                $searchDB['cod_rol_usuario'],
                $searchDB['cod_horario_user'],
                $searchDB['cod_estado_usuario'],
                $searchDB['nombre'],
            );
        }
    }
}

```

```

        $searchDB['apellido'],
        $searchDB['email'],
        $searchDB['contrasenia'],
        $searchDB['question1'],
        $searchDB['question2'],
        $searchDB['question3']
    );

    return $user;

} else {
    return false;
}

} catch (Exception $e) {

    die($e->getMessage());
};
}
}

```

```

<h1 class="tit1">Formulario Registro</h1>

<div class="container-register h-100">

    <form method="POST" action="?c=UserController&a=crearUser" class="form-
flexR" >

        <div >
            <label for="nombre"> Nombre:</label>
            <input id="nombre" name="nombre" type="text" placeholder="Primer
Nombre">
        </div>
        <div >
            <label for="apellido">Apellido:</label><input id="apellido"
name="apellido" class="ml-1" type="text" placeholder="Primer Apellido">
        </div>
        <div >
            <label for="tipoDoc">Tipo-Documento:</label>
            <select name="tipoDoc" class=" w-40">
                <option value="C-N/A">No Asignado</option>

```



```

        <option value="CC">Cedula de Ciudadania</option>
        <option value="RUT">RUT</option>
        <option value="NIT">NIT</option>
        <option value="CE">Cedula de Extranjeria</option>
    </select>
</div>
<div >
    <label for="documento"> Documento:</label> <input
id="documento" name="documento" type="number" placeholder="ingrese el numero
de documento">
</div>
<div >
    <label for="email"> Correo:</label> <input id="email" name="email"
type="email" placeholder="Correo Aqui">
</div>
<div >
    <label for="password"> Contraseña:</label> <input id="password"
name="password" type="password" placeholder="contraseña Aqui">
</div>
<div >
    <label for="telefono">Teléfono:</label> <input id="telefono"
name="telefono" type="number" placeholder=" Número de contacto">
</div>
<div >
    <label for="question1">Pregunta:</label> <input id="question1"
name="question1" type="text" placeholder=" Nombre de pila de su Madre">
</div>
<div class="">
    <label for="question2">Pregunta:</label> <input id="question2"
name="question2" type="text" placeholder=" Nombre de su primer mascota">
</div>
<div class="">
    <label for="question3">Pregunta: </label><input id="question3"
class="ml-1" name="question3" type="text" placeholder=" Nombre de pila de su
Padre">
</div>
<ul class="" style="list-style: none;">
    <li>
        <input style="width: 25px;" id="tyc" name="tyc"
type="checkbox"><a onclick="location.href='compartidos/condiciones.html'"
target="_self">Términos y condiciones</a>
    </li>
    <li>
        <input style="width:25px;" id="td" name="td"
type="checkbox"><a

```

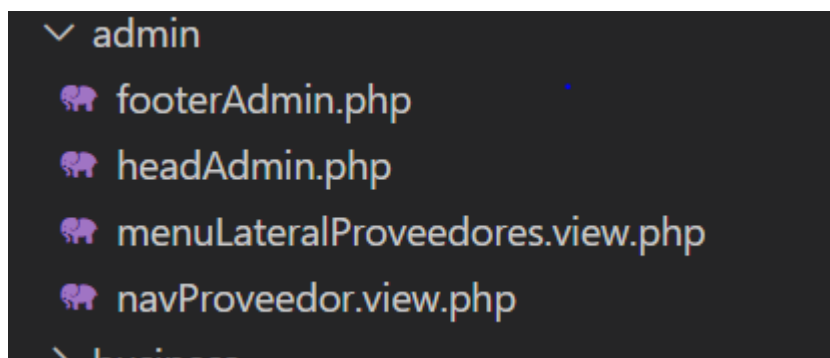
```

onclick="location.href='compartidos/tratamientoData.html'"
target="_self">Tratamiento de datos</a>
    </li>
</ul>
<div>
    <input type="submit" class="form-button col-3"
name="enviarRegistro" value="ENVIAR"></br>
    </div>
</form>
</div>

```

5.- DRY: Dont repeat yourself.

Para hacer modificaciones generales del headear y footer, menuLateralProveedores, navProveedores, estos se encuentran separado para evitar repetir código.



```

</main>
<footer class="footer-Admi">

    <div class="container-fluid align-self-center text-center pt-1">
        <div class="row">
            <div class="col-2">
                Copyright ©
            </div>
            <div class="col-5 container-fluid align-self-center text-
center">
                <h6> Todos los derechos reservados</h6>
            </div>

```

```

    </div>
  </div>

```

```
</footer>
```

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.c
ss" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
  <link href="assests/css/styleGeneralModul.css" rel="stylesheet">
  <link href="assests/css/menuDashboard.css" rel="stylesheet">
  <link href="assests/css/usuarios.css" rel="stylesheet">
  <link href="assests/css/message.css" rel="stylesheet">
  <link href="assests/css/btnUp.css" rel="stylesheet">
  <link href="assests/css/registroProveedor.css" rel="stylesheet">
  <link href="assests/css/tratamientoD.css" rel="stylesheet"
type="text/css" />

  <title>Principal Proveedores</title>
</head>

<body>

  <nav class="navbar navbar-dark" style="background-color: #010014;">

    <div class="container-fluid col-12 justify-content-around"
style="height: 48px; ">
      <a class="navbar-brand " style="margin-left: -50px;">
        <h3>Proveedores</h3>
      </a>
      <div style="margin-top: -10px;margin-left: 90px;">
        <a class='btn btn-success text-white rounded-2 ml-3 col'
href='?c=LogoutController'>SALIR</a>
      </div>
    </div>
  </nav>

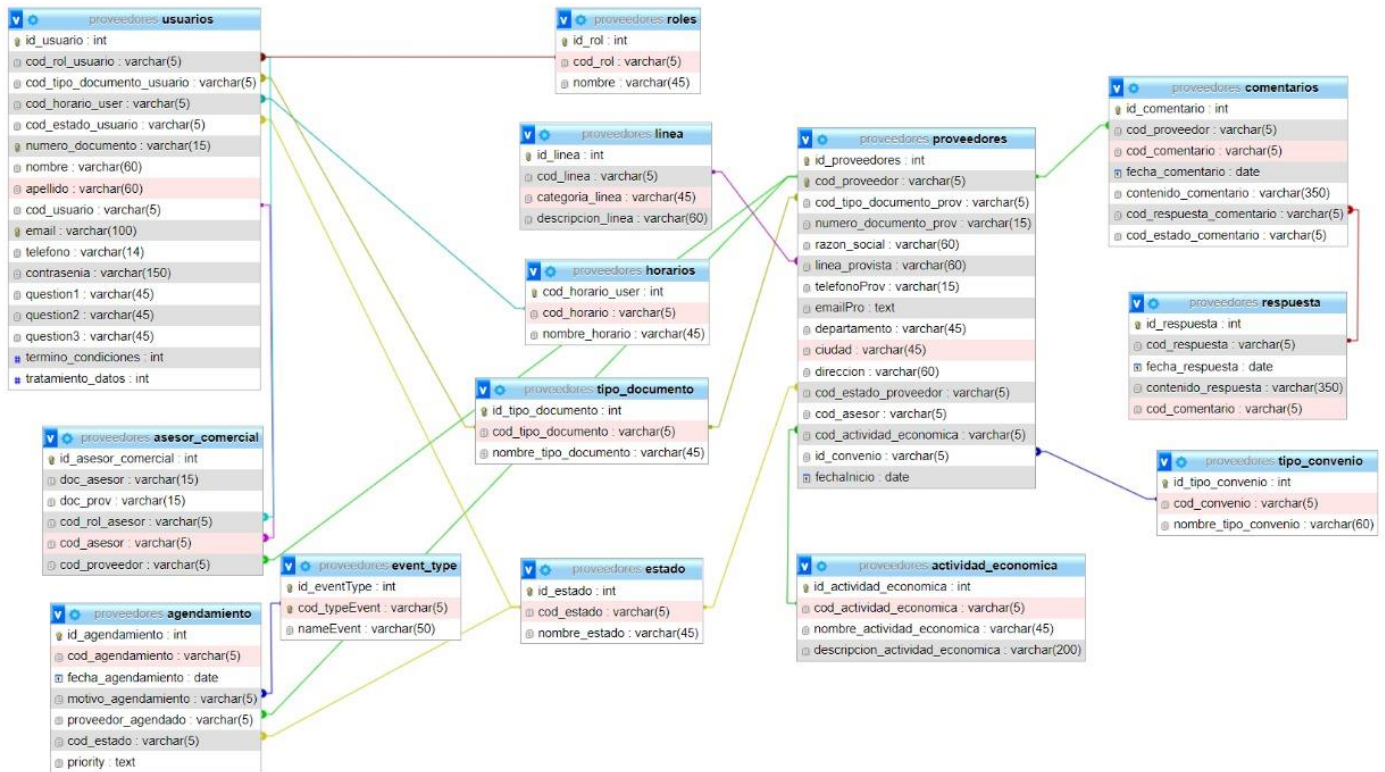
```

```
<main class="mainAdmi">
```

6.- No inventes.

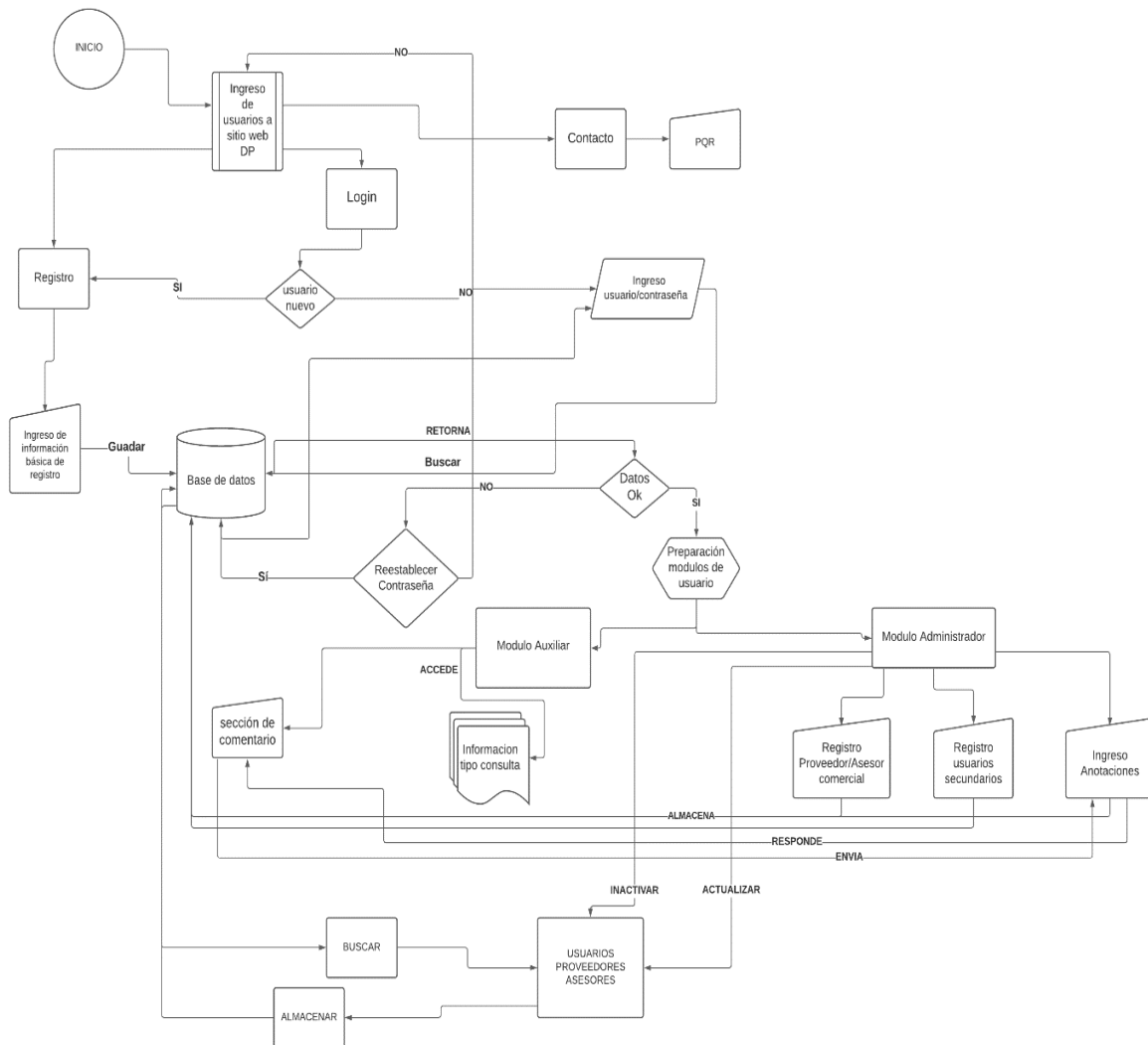
Se cuenta con el api de calendario de Google y phpMailer

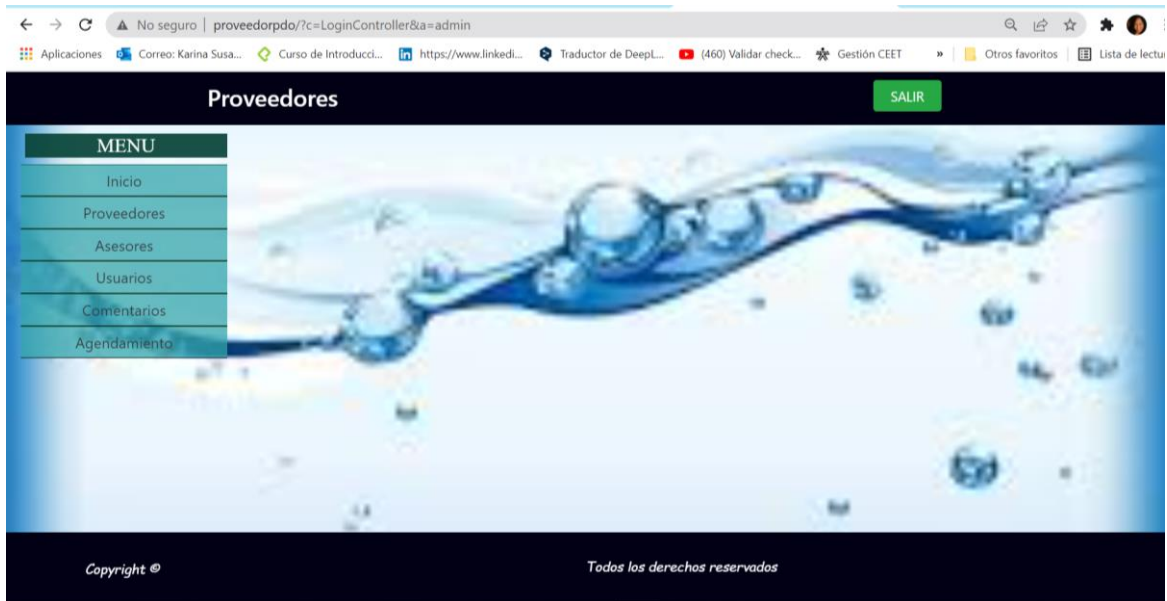
1). La base de datos cuenta con 14 Tablas, se encuentra normalizada en tercera Forma.



Todo el diseño modular: Entradas + Procesos + Salidas

Mediante el modelo lógico, se visualiza que aplicativo web cuenta con 6 Módulos funcionales, 46 páginas de código, 17 css, 1 Script, 20 Interfaces.





Resultados: Impresiones y Transacciones

Resultados de aplicación:

- Informes por pantalla: 3
- Informe Digital:0



↳

Asesores activos									
Empresa	ID Asesor	Nombre	Apellido	2th Apellido	Teléfono	Estado	Acción	Acción	

Módulo Proveedores ^{Auxiliar} [SALIR](#)

[Buscar Proveedor](#)
[Notificaciones](#)
[Consultar Agenda](#)

1. Notificación
2. Notificación
3. Notificación
4. Notificación
5. Notificación
- n. Notificación

Seguridad entre pocos aspectos

El aplicativo cuenta con dos tipos de Rol:

Rol Usuario Administrado

- Registrar
- Actualizar
- Inactivar
- Buscar
- Almacenar

Rol Usuarioaux

- Solo consulta

Para acceder el servicio de proveedores de agendamientos, debe dirigirse a la opción “Ingreso”; esta solicitará que digite el correo y la contraseña con la que realizó su inscripción a la plataforma.



Deployment Programmers Transformación Digital Inicio Nosotros Contáctenos Registrarse Ingreso

Ingreso Usuarios

Usuario

Contraseña

INGRESAR

[Registrarse](#)

[Restablecer Contraseña](#)

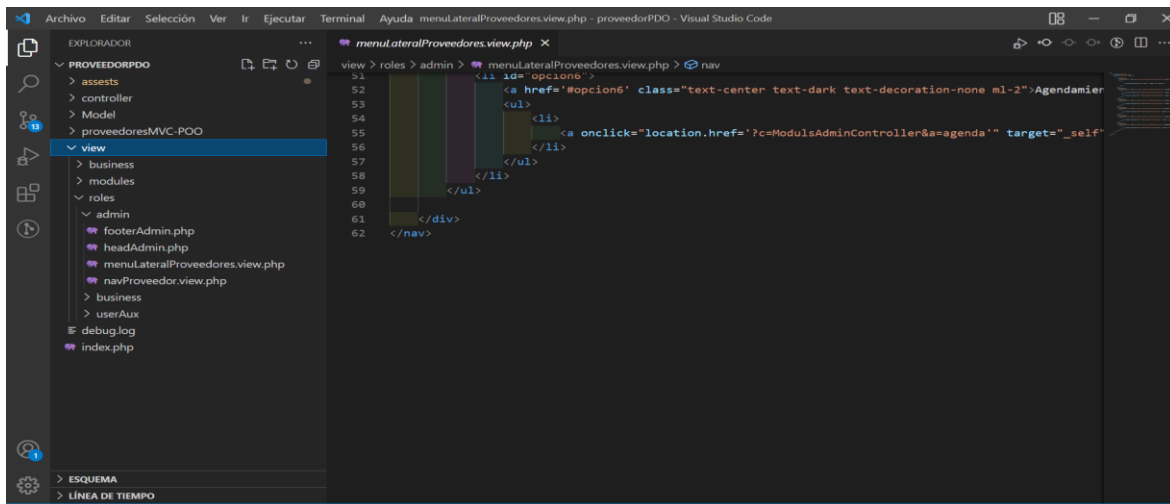
Copyright © Todos los derechos reservados

7.- Comenta tu código.

Para facilitar la modificación y el mantenimiento del aplicativo Web, se trabajó el código teniendo en cuenta las buenas prácticas de desarrollo, el cual permite que otro desarrollador entienda fácilmente el código, adicional se creó manual de usuario y manual técnico.

8.- Divide y vencerás.

El código se divide en modelo, vista y controlador, css, Javascript



9.- Testeo de código.

Durante el desarrollo del aplicativo web, se ha realizado pruebas de caja blanca como son pruebas unitarias entre ellas las pruebas orientada al error y pruebas orientadas al éxito, Pruebas integrales funcionales, se automatizo el formulario de contáctenos.

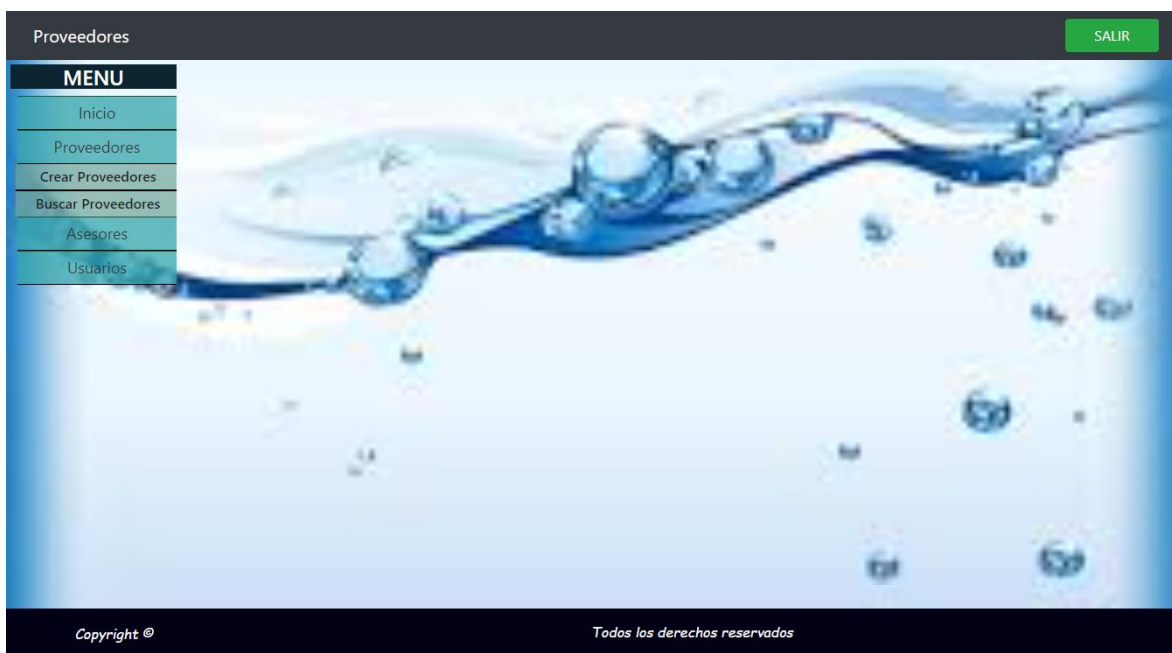
Mensajes de error



CP4. Al digitar los datos de acceso correctamente, re direccionará al usuario a la vista principal de administrador o al principal de auxiliar según sea los permisos de usuario



Primera vista usuario administrador



Primera vista modulo usuario auxiliar

Módulo Proveedores ^{Auxiliar}

Buscar Proveedor Notificaciones Consultar Agenda

Buscar

Razón Social:

Actividad Comercial:

Asesor Comercial :

Telefono Asesor:

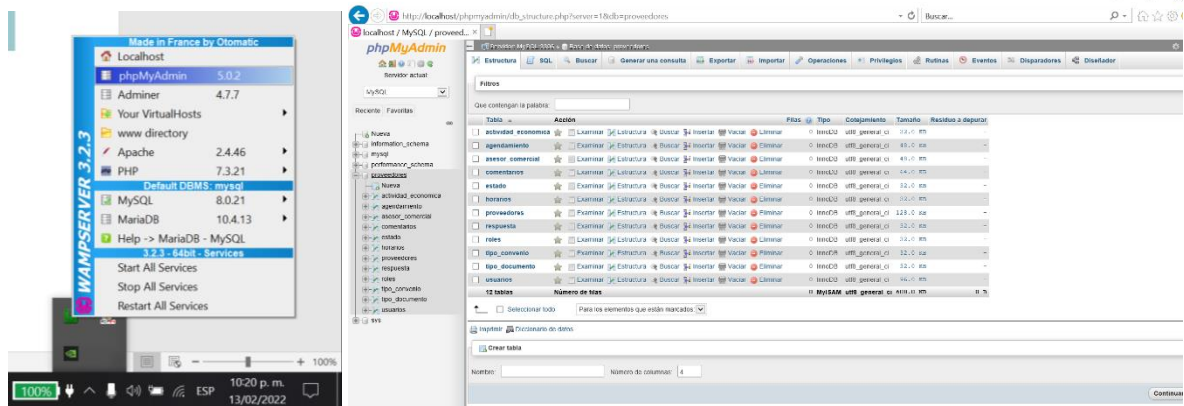
Caja de comentarios

Guardar

Nota: El Plan de prueba de testeo, se encuentra en proceso.

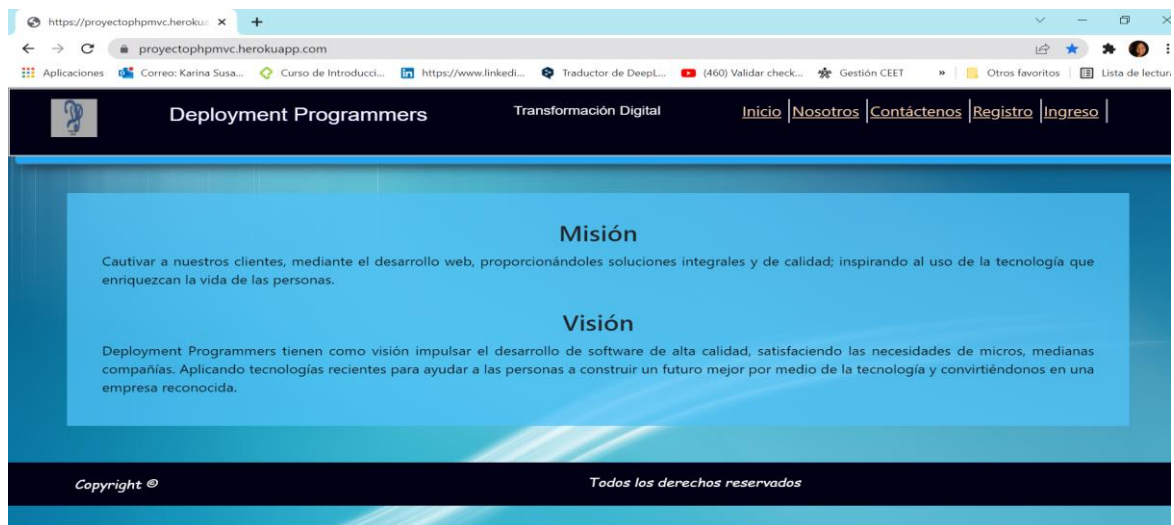
10.- Optimización.

El sistema que asegura los datos es: PhpMyAdmin



Servidor BBDD -> Apache – Wampserver – Heroku

<https://proyectophpmvc.herokuapp.com/>



Deployment Programmers



Mysql - Hosting Gratuito con CleverCloud

The screenshot shows the Clever Cloud console with a MySQL instance named 'MySQL Dev' (ID: mysql_be36dd8c-c5e9-4b2f-88af-8ab7adc08fe6) in the 'ACTIVE' state. The instance is running MySQL 8.0 on a 'par-mysql-c6' plan. The phpMyAdmin interface is visible, showing a database named 'bdvrptxv379z0mfxlqfj' with a 'New' button and a list of tables including 'ACTIVIDAD_ECONOMICA', 'AGENDAMIENTO', 'ASESOR_COMERCIAL', 'COMENTARIOS', 'ESTADO', 'HORARIOS', 'PROVEEDORES', 'RESPUESTA', 'ROLES', 'TIPO_CONVENIO', 'TIPO_DOCUMENTO', and 'USUARIOS'. The console also shows a sidebar with 'Personal space' and 'Add dashboard' options.

<https://desktop.github.com/>

The screenshot shows the GitHub Desktop application interface. The 'Branches' panel is open, displaying a list of branches: 'main' (16 hours ago), 'Karina' (16 hours ago, selected), 'develop' (16 hours ago), 'origin/YeceniaDevelop' (16 hours ago), and 'origin/jesudp' (15 days ago). The 'Publish branch' button is visible at the top right. The background shows the 'Changes' panel with a 'Commit to Karina' button and a 'Summary (required)' field. A notification at the bottom indicates that the screen is being shared on a meeting.

Algunas métricas que se tendrán en cuenta al momento de realizar pruebas.

Política de calidad del proyecto:

Este proyecto debe cumplir con los requisitos desde el punto de vista de los pequeños comerciantes, es decir acabar dentro del tiempo proyectado de estudio y también debe cumplir con los requisitos de calidad del software.

Factor de calidad relevante	Objetivo de calidad	Métrica a utilizar	Frecuencia y momento de medición	Frecuencia y momento de reporte
Errores del proyecto	$EC \geq 0.95$	EC = Número de errores HTML encontrados/ Número de errores permitidos	Frecuencia semanal	Frecuencia semanal
Desempeño Cronograma	$SPI \geq 0.95$	SPI= ejecución actual del proyecto/Ejecución planeada	Frecuencia semanal	Frecuencia semanal
Eficiencia	$EF \geq 98\%$	Segundo en ejecutar una acción /sobre segundo esperados	Frecuencia semanal	Frecuencia semanal
Prueba de software	$PS \geq 95\%$	Número de pruebas realizadas / Números de pruebas requeridas para tener una cobertura adecuada	Frecuencia semanal	Frecuencia semanal
Desempeño de costo	$CPI \geq 0.95$	Costo actual del proyecto/Costo planeado	Frecuencia semanal	Frecuencia semanal
Funcionalidad	$F \geq 0.95$	No. De requerimiento implementados/ No. De requerimientos planeados	Final del proyecto	Final del proyecto

Plan de Mejora de Procesos:

1. Define los objetivos.
2. Mide y analiza los resultados.
3. Comparativa de resultados entre procesos.
4. Simplifica procesos.
5. Involucrar al equipo de trabajo.
6. Evaluación periódica del **plan de mejora continua**.
7. Análisis y feedback de los resultados.

12.- Documentación.

Matriz de actividades de Calidad

Entregables	Estándares de calidad aplicable	Actividades de prevención	Actividades de control	Estado
Fase Análisis del sistema		Revisión estándar	Revisión /Aprobación miembros del proyecto	Realizado
Informe de requerimientos	Norma IEEE 830	Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso
Diagrama Gantt Proyecto		Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso
Calidad del software	Norma ISO 9126	Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso
Fase de diseño		Revisión estándar	Revisión /Aprobación miembros del proyecto	Realizado
Diagrama Caso de Uso	Estándar UML 2.0	Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso
Diagrama de Clases	Estándar UML 2.0	Revisión estándar	Revisión /Aprobación miembros del proyecto	Realizado
Diagrama de secuencias	Estándar UML 2.0	Revisión estándar	Revisión /Aprobación miembros del proyecto	Pendiente
Diagrama de distribución	Estándar UML 2.0	Revisión estándar	Revisión /Aprobación miembros del proyecto	Realizado
Diagrama Entidad - Relación Base de Datos	Notación Crows foot	Revisión estándar	Revisión /Aprobación	Realizado

			miembros del proyecto	
Mockup del sistema	Norma ISO 9126	Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso
Arquitectura del sistema	Norma ISO 9126	Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso
Fase de Implementación y Desarrollo		Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso
Diseño de Interfaces	Norma ISO 9126	Revisión estándar	Revisión /Aprobación miembros del proyecto	Realizado
Codificación del sistema	Norma ISO 9126	Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso
Fase prueba integración		Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso
Manual de Usuario Y técnico	Norma ISO 9126	Revisión estándar	Revisión /Aprobación miembros del proyecto	En curso

Modelo de Calidad aplicado -McCall

A continuación, se describen las seis características para la evaluación de la calidad externa y su descomposición en subcaracterísticas, como se explican en ISO 9126-1

Operación del producto

Factores de calidad	Descripción del factor y criterios de evaluación de la calidad
---------------------	--

Corrección	Cumple con el objetivo para el cual fue creado.
Fiabilidad	El desarrollo del aplicativo Web, permite que el usuario reconozca fácilmente la estructura lógica del sistema.
Eficiencia	El aplicativo web cuenta con un manual, que facilita la comprensión del uso del mismo.
Integridad	El aplicativo web, permite interactuar al usuario entre los diferentes módulos (registro, Ingreso de usuario, módulo administrador, módulo usuario auxiliar).
Usabilidad	Los textos y mensajes son claros, sin ambigüedades de tal manera que son entendibles para un usuario con un conocimiento básico en el uso de la computadora.

Revisión del producto

Factores de calidad	Descripción del factor y criterios de evaluación de la calidad
Facilidad de evaluación	Los niveles de madurez del aplicativo web, los podemos evaluar, mediante los logros de los objetivos genéricos y específicos que se aplica a cada proceso.
Flexibilidad	El desarrollo del aplicativo web, se considera tolerante al fallo. Teniendo en cuenta que, si el sistema presentara demora en el tiempo de respuesta, no significa que los módulos dejen de funcionar.

Transición del producto

Factores de calidad	Descripción del factor y criterios de evaluación de la calidad
Portabilidad	El aplicativo no cuenta con la portabilidad, debido a que no es responsive
Reusabilidad	Para el módulo de agendamiento, se consume la Appi de Calendario de Google y para el módulo de contáctenos se utiliza Php Mailer.
Interoperabilidad	El aplicativo cumple con este criterio, teniendo en cuenta que el aplicativo puede comunicarse con la base de datos de Mysql, las appi de Calendario de Google y php Mailer



OPERACIÓN

Corrección:

¿El software hace lo que necesito?

Mediante encuesta al usuario, se evalúa el grado de satisfacción del servicio que se brinda a través del aplicativo web, adicional el mismo cuenta con el módulo de contáctenos el cual le permite al usuario reportar: las quejas, peticiones o reclamos con el producto.

Fiabilidad:

¿Lo Hace de forma exacta siempre?

El sistema soporta una disponibilidad de sus servidores de 24x7 al 98% de disponibilidad.

Eficiencia:

¿Aprovecha el hardware lo mejor posible?

Integridad:

¿Es seguro?

Los usuarios tienen acceso a las opciones necesarias para el cumplimiento de sus funciones, según el rol del usuario.

Facilidad de Uso:

¿Lo puedo usar con facilidad?

REVISION

Facilidad de prueba:

¿Se puede probar?

Flexibilidad:

¿Se puede modificar?

Facilidad de mantenimiento:

¿Qué tan difícil es arreglarlo?

TRANSICION

Portabilidad:

¿Se puede usar en otra máquina?

Reusabilidad:

¿Se puede reutilizar parte del Software?

Deployment Programmers



Interoperabilidad:

¿Se puede comunicar con otros sistemas?