1. INTRODUCCION

Se desarrollará un servicio web que ofrezca la opción a usuarios cliente el poder presentar una queja o reclamo ante una entidad educativa con el fin de poder ser resuelta y así aclarar inquietudes.
Se planea desarrollar este servicio con los conocimientos adquiridos en clase como lo es la programación y la lógica computacional, llevándolos a cabo mediante código y plasmando en ellos nuestra capacidad de desarrolladores para culminar este proyecto que nos ampliará aún más nuestras aptitudes a nivel profesional que se verán reflejadas en un futuro en el ámbito laboral.

Las herramientas que se usarán será un servidor web (XAMPP), un motor de base datos (MySQL), un editor de código (VSC), esto con el fin de realizar el proyecto lo más óptimo y organizado posible ya que se pretende entregar un servicio web funcional con todas las opciones y campos que necesitan los usuarios y así cumpliendo con los requerimientos plasmados para su realización.

PQRS

Ingeniería de Sistemas
Universidad de Investigación y Desarrollo
2021

Estructura y contenido

2. Definiciones y especificación de requerimientos

El sistema ofrecerá un campo de peticiones, quejas, reclamos o sugerencias el cual permitirá dar a conocer las inquietudes y manifestaciones que tienen nuestros grupos de interés, teniendo así distintos perfiles los cuales van a manejar diferentes tareas ya sea de rol común, gerente o administrador.

a) ***Definición general del proyecto de software:*** Este sistema consiste en que cualquier persona que tenga una petición podrá hacerla en nuestra plataforma para darla a conocer específicamente al ente que al que va dirigido y estará controlado por nuestros administradores para poderlas contestar o redirigirlas respectivamente.
b) ***Especificación de requerimientos del proyecto:*** El sistema contará con un apartado de texto libre en el que la persona podrá escribir su queja con un número de caracteres limitado el cual llegará a manos de nuestros usuarios. Estos también están limitados respecto a su nivel; El nivel 1 solo podrá contestar a las peticiones solo si estas van dirigidas a él. El nivel 2 también podrá contestarlas pero a su vez podrá ver todas las peticiones entrantes al sistema. Por último tenemos al nivel 3 que es el nivel de administrador el cual contará con todo el menú completo en el que podrá responder las peticiones, ver los últimos cambios y además podrá redirigirlas al campo en específico.
c) ***Procedimientos de instalación y prueba:*** Este sistema se realiza en los lenguajes de HTML, php y JavaScript ya que este es un software que va dirigido a la web y se ejecutará en un ámbito educativo.

De la definición general del proyecto de software

- Idea general: Software PQRS
- Objetivo general: Desarrollar un software orientado a la web para la gestión de las pqrs del Instituto Técnico en Comunicaciones de Barrancabermeja – Santander.
- Objetivo específico 1: Elaborar un mapa de procesos mediante diagramas de flujos funcionales y el diagrama para el modelo relacional de la base de datos.
- Objetivo específico 2: Desarrollar el software utilizando el patrón MVC y la metodología SCRUM.
- Objetivo específico 3: Elaborar un manual de usuario describiendo todas las funcionalidades y roles existentes en el software.
- Usuarios: El software debe reconocer 4 tipos de usuarios que serán diferenciados según su nivel para definir su categoría

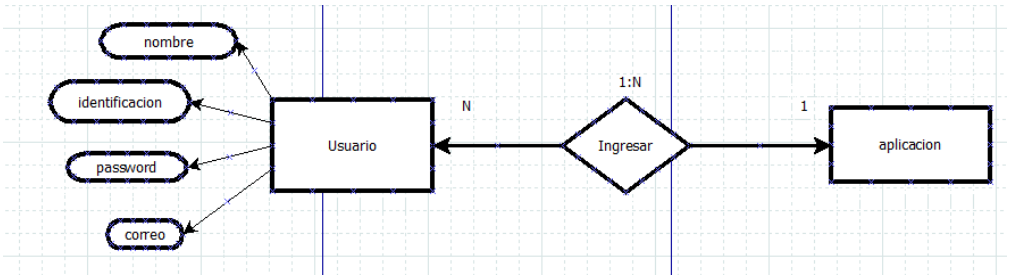
De las especificaciones de procedimientos

Procedimientos de desarrollo:

- Herramientas utilizadas: XAMP, Visual Studio Code, HTML, Php, js, css, HeidiSQL.
- Planificación: Se verá una página principal para el usuario común y otra para los usuarios registrados como lo son los gerentes y los administradores. Estas páginas estarán programadas con HTML y CSS (bootstrap) para la parte estética y con php en su funcionalidad donde estará conectado a una base de datos para llevar todos los registros.

3. Arquitectura del sistema

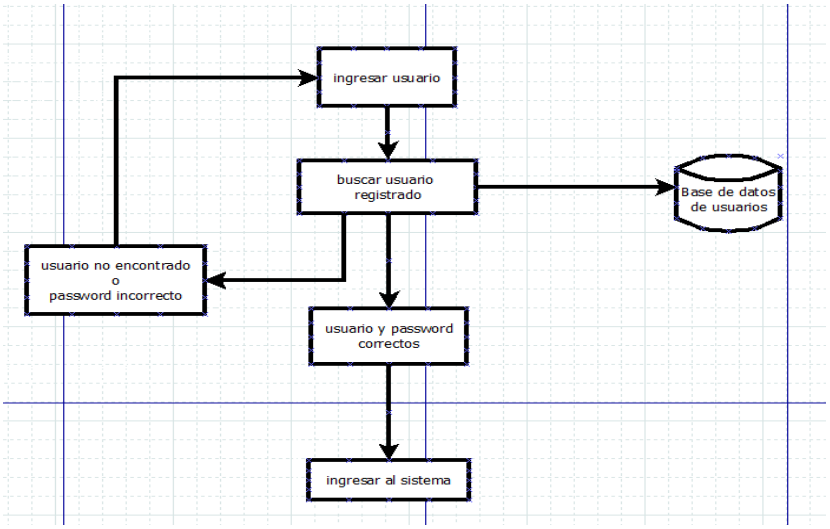
- **Descripción jerárquica:** Los componentes de este sistema están organizados por paquetes de tal manera que encuentren fácilmente la ruta correcta para comunicarse entre sí.
- **Diagrama de módulos:**



- **Descripción individual de los módulos:**

  - **Descripción general y propósito:** El módulo ingresar valida la información del usuario; El modulo usuario valida el nombre, la identificación, el password y el correo del usuario. Una vez validado esto, la aplicación permite el ingreso.
  - **Dependencias:** La aplicación depende del módulo ingresar; El módulo ingresar depende del módulo usuario y hasta que este no valide todos los campos no dará luz verde para ingresar a la aplicación, es como un efecto cadena.

4. Diseño del modelo de datos

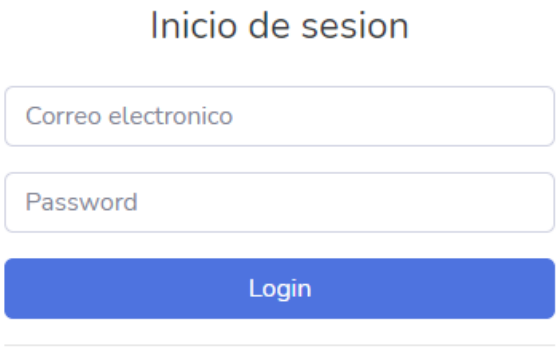


5. METODOLOGÍA

-Login

Lo primero que se realizó fue el login con el que los usuarios van a poder ingresar a la plataforma y realizar sus tareas dependiendo del nivel que tengan. Esto se logró usando el lenguaje de HTML y creando sus apartados para ingresar el correo y la contraseña del usuario

```
<div class="form-group">
    <input type="email" class="form-control form-control-user"
        id="email" name ="email" aria-describedby="emailHelp"
        placeholder="Correo electronico">
</div>
<div class="form-group">
    <input type="password" class="form-control form-control-user"
        id="password" name="password" placeholder="Password">
</div>
```



Y su respectivo botón para ingresar. En el apartado del diseño se agregaron librerías del css más específicamente de bootstrap para que visualmente se viera más organizado el login

También a la hora de iniciar la sesión, protegimos las contraseñas con argón 2 y esto lo hicimos mediante un hasheado con el lenguaje de PHP

```php
<?php

    $password = '123456';

    echo password_hash($password,PASSWORD_ARGON2I);

?>
```

```php
$passwordHash = password_hash($password,PASSWORD_ARGON2I);
```

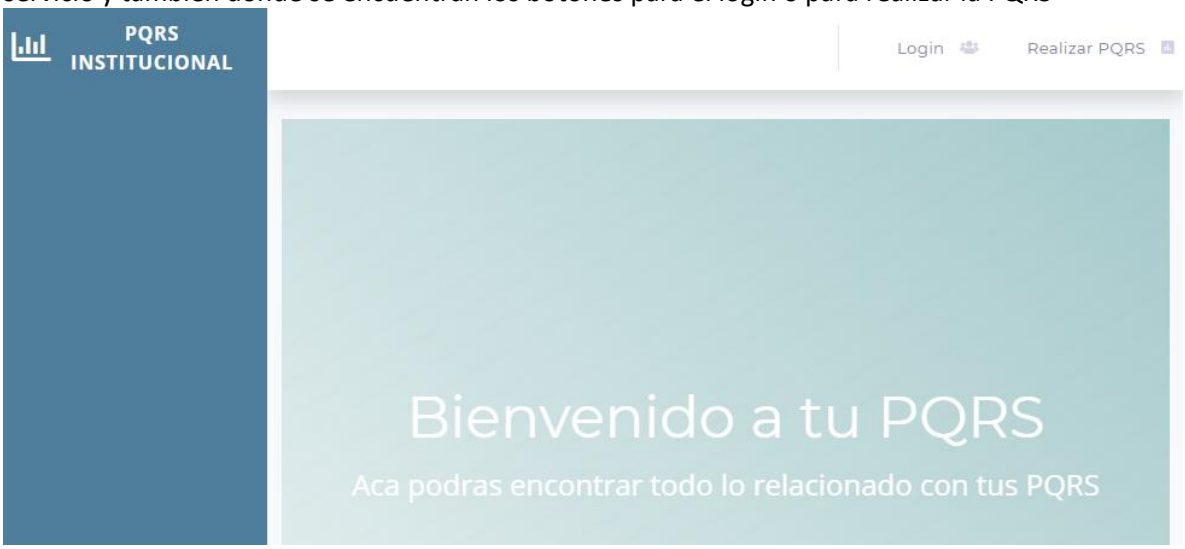
-Inicio

Lo segundo en realizarse fue la página de inicio, en esta página se da una introducción al usuario cliente del que y como funciona una PQRS y cuál es su importancia. Los lenguajes usados fueron un poco de PHP para la parte lógica y HTML para complementar el diseño. Con PHP se incluyó el templateinicio, llamándolo así mediante el código que se ve en la figura

Inicio.php

```php
<?php
    include_once('templateInicio.php');
?>
```

En el templateInicio es donde está escrito todo el código que se muestra en la página principal del servicio y también donde se encuentran los botones para el login o para realizar la PQRS



```
<div class="topbar-divider d-none d-sm-block"></div>

<li class="nav-item dropdown no-arrow">
    <a class="nav-link dropdown-toggle" href="login.html" id="userDropdown" role="button">
        <span class="mr-2 d-none d-lg-inline text-gray-600 small">Login</span>
        <i class="fas fa-users" href="login.html"></i>
    </a>
</li>
<li class="nav-item dropdown no-arrow">
    <a class="nav-link dropdown-toggle" href="ingresarpqrs.php" id="userDropdown" role="button">
        <span class="mr-2 d-none d-lg-inline text-gray-600 small">Realizar PQRS</span>
        <i class="fas fa-poll" href="ingresarpqrs.php"></i>
    </a>
 </li>
```

En este apartado también se agregó el favicon de la página que es el logo con el que se identifica, aparte de las librerías de css, para terminar con HTML se realizó todo el body de la página usando sus respectivos div para tener todo organizado y eligiendo la paleta de colores y las imágenes ilustrativas que estarán en esta página de inicio.

-Registrar Usuario

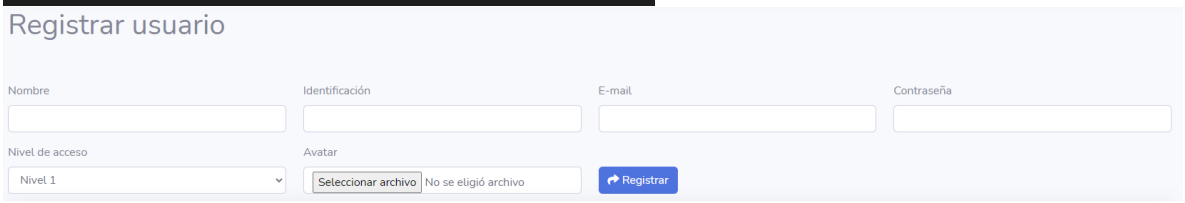
Aquí se hizo el apartado para registrar los usuarios a la base de datos que serán los que dividiremos por nivel según su rango. Para esto lo que hicimos fue llamar a la base de datos usando el siguiente código

```
<?php
    error_reporting(E_ALL);
    ini_set('display_errors', '1');

    include_once('Controllers\tablaUsuarios_controller.php');
    include_once('template.php');

    $PQRS = new queries();

    $UsuariosT = $PQRS->getUsuariosT();
?>
```



Aquí también se llamó un controlador de la tabla de usuarios que es el encargado de llevar el registro de los usuarios que se encuentran la base de datos y nos ayuda a validarlos con la siguiente función

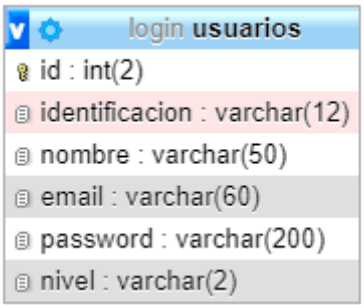
```
function getUsuariosT(){
    $conexionLogin = new conexionLogin();
    $conexion = $conexionLogin->conectar();

    $sql = $conexion->prepare('
    SELECT * FROM usuarios
    ');
```

También encontramos el apartado en donde clasificamos a estos usuarios por su rol, esto lo logramos definiendo las variables con números enteros en donde 1 será el líder de área, el 2 será el moderador y el 3 será el administrador

```
<div class="col-md-3">
    <label for="level" class="form-label">Nivel de acceso</label>
    <select class="form-control" name="level" id="level">
        <option value="1">Nivel 1</option>
        <option value="2">Nivel 2</option>
        <option value="3">Nivel 3</option>
    </select>
</div>
```

Y así se vería en la base de datos



-Ingresar PQRS

También tenemos la parte en donde se ingresan y se registran las PQRS de los usuarios clientes llamando a las áreas para saber a dónde va dirigido

```
<?php

    include_once('templateInicio.php');
    include_once('Controllers\tablaUsuarios_controller.php');

    $PQRS = new queries();

    $Areas = $PQRS->getAreas();
?>
```

Y a su vez realizando los campos a llenar con lenguaje HTML donde nos pedirá lo siguiente

-Datos básicos del emisor
    -Nombre completo
    -Número de identificación
    -Teléfono
    -Email (opcional)
-Área de envío
-Asunto
-Descripción
-Adjuntos

```
<div class="row">
   <div class="col-md-3 form group">
      <label>Nombre Completo</label>
      <input type="text" class="form-control" id="nombre" name="nombre" required>
   </div>
   <div class="col-md-3 form group">
      <label>Identificacion</label>
      <input type="number" class="form-control" id="identificacion" name="identificacion" required>
   </div>
   <div class="col-md-3 form group">
    <label>Telefono</label>
    <input type="number" class="form-control" id="telefono" name="telefono" required>
   </div>
   <div class="col-md-3 form group">
    <label>Email</label>
    <input type="email" class="form-control" id="email" name="email">
   </div>
   </div><br><br>
   <!-- datos -->
   <div class="row">
      <div class="col-md-4 form-group">
         <label for="">Area destino</label>
         <select class="form-control" name="area" required>
         <?php
                  foreach ($Areas as $key) {
         ?>
            <option value="<?=$key['nombreArea']?>"><?=$key['nombreArea']?></option>
          <?php
                  }
          ?>
         </select>
      </div>
      <div class="col-md-4 form-group">
         <label for="">Asunto</label>
         <input type="text" class="form-control" id="asunto" name="asunto" required>
      </div>
      <div class="col-md-4">
      <label for="adjunto">Adjunto</label>
      <input type="file" class="form-control" name="adjunto" id="adjunto">
      </div>
   </div>

      <div class="row">
         <div class="form-group">
         <label for="">Descripcion</label>
         <textarea class="form-control" id="descripcion" name="descripcion" rows="3" required></textarea>
      </div>
```

-Conexión login

En esta parte encontramos la base de datos como tal que está registrada en el phpmyadmin y es la que recibe todos los datos de los usuarios y de las pqrs que entran

```
<?php

   class conexionLogin{
      private $host = 'localhost';
      private $dataBase = 'login';
      private $usuario = 'DeyberS';
      private $password = 'Bohorquez16';
      private $atributos = array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION, PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,);

      protected $conexion;

      public function conectar(){

         try{
            $this->conexion = new PDO("mysql:host={$this->host};dbname={$this->dataBase};charset=utf8", $this->usuario, $this->password, $this->atributos);
            return $this->conexion;
         }catch(PDOException $e) {
            echo 'Error conectando con la base de datos: ' . $e->getMessage();
         }

      }

      public function desconectar(){
         $this->conexion = null;
      }

   }
```

-Registrar usuario controller

Aquí es donde se encuentra el apartado de crear el usuario como tal, estarán los campos para llenar los datos

```php
<?php

    error_reporting(E_ALL);
    ini_set('display_errors', '1');
    include_once('../dataBase/conexionLogin.php');
    include_once('seguridad_controller.php');

    $conexionLogin = new conexionLogin();
    $conexion = $conexionLogin->conectar();

    $identificacion = $_POST['identificacion'];
    $nombre = $_POST['nombre'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $level = $_POST['level'];

    $response = array();
```

Se usó el método POST para estos campos ya que nos permite enviar información al servidor, recordemos que solo los usuarios de rol administrador podrán crear usuarios nuevos por eso es importante definir el nivel del nuevo usuario.

-Registrar PQRS controller

También tenemos un controlador para las PQRS que entran en donde primero se hará una conexión con el login para recibir los datos

```php
<?php

    error_reporting(E_ALL);
    ini_set('display_errors', '1');
    include_once('../dataBase/conexionLogin.php');
    include_once('seguridad_controller.php');

    $conexionLogin = new conexionLogin();
    $conexion = $conexionLogin->conectar();

    $nombre = $_POST['nombre'];
    $identificacion = $_POST['identificacion'];
    $telefono = $_POST['telefono'];
    $email = $_POST['email'];
    $area = $_POST['area'];
    $asunto = $_POST['asunto'];
    $descripcion = $_POST['descripcion'];

    $response = array();


    $path = "img/adjuntos/";
```

Una vez recibidos los datos, se procederá a validarlos haciendo primero una conexión a la base de datos para corroborar que todos los datos estén digitados correctamente

```php
$conexion->beginTransaction();

$sql = $conexion->prepare("
    INSERT INTO pqrs (nombre, identificacion, telefono, email, area, asunto, adjunto, descripcion)
    VALUES (:nombre, :identificacion, :telefono, :email, :area, :asunto, :adjunto, :descripcion)
");
$sql->bindParam(':nombre', $nombre);
$sql->bindParam(':identificacion', $identificacion);
$sql->bindParam(':telefono', $telefono);
$sql->bindParam(':email', $email);
$sql->bindParam(':area', $area);
$sql->bindParam(':asunto', $asunto);
$sql->bindParam(':adjunto', $adjunto);
$sql->bindParam(':descripcion', $descripcion);
$sql->execute();
```

Una vez se validen los datos y estén correctos se envía la información con un “true” que arrojará un mensaje confirmando el envió de la PQRS

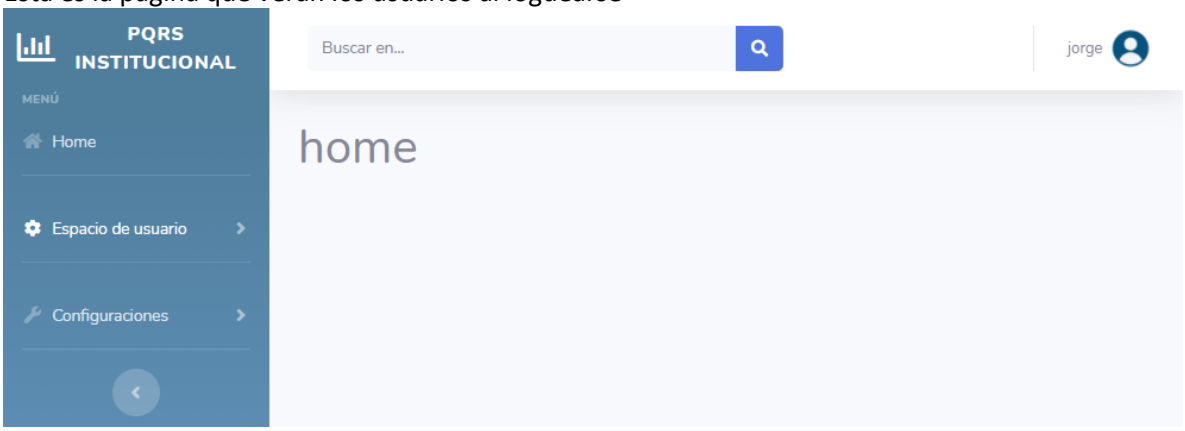
```php
$conexion->commit();
$response['estado'] = true;
$response['mensaje'] = 'PQRS Registrada!';
```

Si no es correcto se negará el envío con un “false”

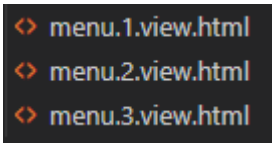
```php
} catch (\PDOException $th) {
    $conexion->rollBack();
    $response['estado'] = false;
    $response['mensaje'] = 'PQRS no Registrado. Error: $th';
    exit();
}
```

-Home

Esta es la página que verán los usuarios al loguearse



Claramente este home cambiará dependiendo del nivel que tenga el usuario y esto se logra creando cada una de las vistas con el lenguaje de HTML



Donde el menú 1 solo tendrá las opciones de ver las PQRS que le lleguen a su área y poderlas responder

```
<div class="bg-white py-2 collapse-inner rounded">
    <h6 class="collapse-header">Opciones</h6>
    <a class="collapse-item" href="estadisticas.php">Estadisticas</a>
    <a class="collapse-item" href="pqrs.php">Pqrs resividad</a>
    <a class="collapse-item" href="404.php">error 404!!</a>
</div>
```

En el menú 2 se verán estas mismas opciones más la de mover el área de la PQRS

```
<div id="collapseUtilities" class="collapse" aria-labelledby="headingUtilities"
    data-parent="#accordionSidebar">
    <div class="bg-white py-2 collapse-inner rounded">
        <!-- <h6 class="collapse-header">Custom Utilities:</h6> -->
        <!-- <a class="collapse-item" href="utilities-color.html">Colors</a>
        <a class="collapse-item" href="utilities-border.html">Borders</a>
        <a class="collapse-item" href="utilities-animation.html">Animations</a> -->
        <a class="collapse-item" href="registrarUsuario.php">Mover area de PQRS</a>

    </div>
</div>
```

Y el menú 3 tiene todas las opciones más la de agregar nuevos usuarios y nuevas áreas

```
<div class="bg-white py-2 collapse-inner rounded">
    <!-- <h6 class="collapse-header">Custom Utilities:</h6> -->
    <!-- <a class="collapse-item" href="utilities-color.html">Colors</a>
    <a class="collapse-item" href="utilities-border.html">Borders</a>
    <a class="collapse-item" href="utilities-animation.html">Animations</a> -->
    <a class="collapse-item" href="registrarUsuario.php">Registrar usuario</a>
    <a class="collapse-item" href="crearAreas.php">Registrar Area</a>

</div>
```

-Menú controller

Para poder llevar a cabo el paso anterior es necesario tener un controlador que es el encargado de asignar cada uno de estos menús según el nivel del usuario y se realiza de la siguiente manera

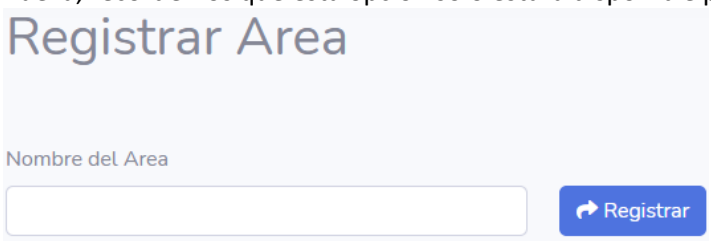
```php
<?php
    error_reporting(E_ALL);
    ini_set('display_errors', '1');

    function getLevel($level){
        $filename = 'views/menu.'.$level.'.view.html';
        return getMenu($filename);
    }

    function getMenu($filename){
        if(is_file($filename)){
            ob_start();
            include($filename);
            return ob_get_clean();
        }
        return false;
    }
```

-Crear áreas

En este apartado se encuentra el código correspondiente a la parte visual para registrar un área nueva, recordemos que esta opción solo estará disponible para el usuario administrador




```php
<h1>Registrar Area</h1>

<form id="formCrearArea" enctype="multipart/form-data">
    <div class="row form-group pt-5">
        <div class="col-md-3">
            <label for="nombreArea" class="form-label">Nombre del Area</label>
            <input type="text" class="form-control" id="nombreArea" name="nombreArea" required>
        </div>

        <div class="col-md-3" style="padding-top: 2em;">
            <button type="submit" class="btn btn-primary"><i class="fas fa-share" onclick="registrarArea()"></i> Registrar</button>
        </div>
    </div>
</form>
```

-Crear área controller

Este campo también tiene su propio controlador que cumple la función de registrar dichas áreas a la base de datos haciendo su respectiva validación y conexión con la misma

```php
<?php

    error_reporting(E_ALL);
    ini_set('display_errors', '1');
    include_once('../dataBase/conexionLogin.php');
    include_once('seguridad_controller.php');

    $conexionLogin = new conexionLogin();
    $conexion = $conexionLogin->conectar();

    $nombreArea = $_POST['nombreArea'];
```

De esta forma es como se validan los datos

```
$conexion->beginTransaction();

$sql = $conexion->prepare("
    INSERT INTO areas (nombreArea)
    VALUES (:nombreArea)
```

Si todo es correcto se registrará el nuevo área

```
$sql->bindParam(':nombreArea', $nombreArea);
$sql->execute();

$conexion->commit();
$response['estado'] = true;
$response['mensaje'] = 'Area Registrada!';
```

Si no, se dará el aviso que no fue registrada

```
} catch (\PDOException $th) {
    $conexion->rollBack();
    $response['estado'] = false;
    $response['mensaje'] = 'Area no Registrada. Error: $th';
    exit();
}
```

6. CONCLUSION

Con la realización de este proyecto nuestros conocimientos se ampliaron tanto que nos sentimos capacitados para la realización de diferentes servicios orientados a la web, adquiriendo nuevos conceptos a lo largo del desarrollo del código que antes no teníamos, pudiendo así cumplir con todos los requerimientos necesarios para el correcto funcionamiento de este software. También resultó muy interesante ya que descubrimos funciones de los lenguajes manejados que no teníamos idea que existían y con esto se despliega todo un abanico de posibilidades a la hora de programar.

Descubrimos que tan importante es una PQRS para ayudar a mejorar una organización y más importante aún es su desarrollo, que nos brinde los campos necesarios y correspondientes para que el usuario pueda diligenciar su inquietud o sugerencia, pudiendo así que este programa esté en un óptimo estado para que los líderes de área, los moderadores y los administradores puedan hacer su respectivo uso y dar solución a los problemas. También ayuda en gran medida a llevar un orden de las estadísticas y los reportes que se van generando en este caso para la institución.

Se realizó un excelente trabajo en conjunto de 3 desarrolladores para llevar a cabo este proyecto y se cumplió con todos los objetivos y requerimientos. El programa se entregó óptimo y funcional para su uso pero no fue sencillo, presentamos diferentes dificultades pero se logró dar solución a cada obstáculo ya que nuestro objetivo era culminar este proyecto y dejarlo funcionando correctamente.

“**Trabajar en equipo** divide el **trabajo** y multiplica los resultados”