

Proyecto Integrador

Software de PQRS
HolaMundo!

PQRS

Introducción

Desarrollo de un software orientado a la web para la gestión de las pqrs del Instituto Técnico en Comunicaciones de Barrancabermeja - Santander

1. 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.

Objetivo General

Desarrollar un software orientado a la web para la gestión de las pqr's del Instituto Técnico en Comunicaciones de Barrancabermeja - Santander

Objetivo Especifico 1

Elaborar un mapa de procesos mediante diagramas de flujos funcionales y el diagrama para el modelo relacional de la base de datos.

Objetivo Especifico 2

Desarrollar el software utilizando el patrón MVC y la metodología SCRUM.

Objetivo Especifico 3

Elaborar un manual de usuario describiendo todas las funcionalidades y roles existentes en el software.

Tipos de usuarios

El software debe reconocer 4 tipos de usuarios que serán diferenciados según su nivel para definir su categoría

login usuarios	
id	int(2)
identificacion	varchar(12)
nombre	varchar(50)
email	varchar(60)
password	varchar(200)
nivel	varchar(2)

Usuario Cliente

Este usuario no tiene poder alguno, solo podrá crear una PQRS adjuntando algunos datos

Usuario líder de área

Usuario que recibe la PQRS solo si está dirigido a su área y la podrá responder

Usuario moderador

Este usuario podrá redirigir las PQRS hacia algún área en específico y a su vez podrá responder dichas PQRS

Usuario administrador

El usuario administrador tiene el control total del software teniendo así los mismo poderes que un moderador pero este si podrá agregar nuevas áreas y nuevos usuarios. No podrá eliminar PQRS o usuarios.

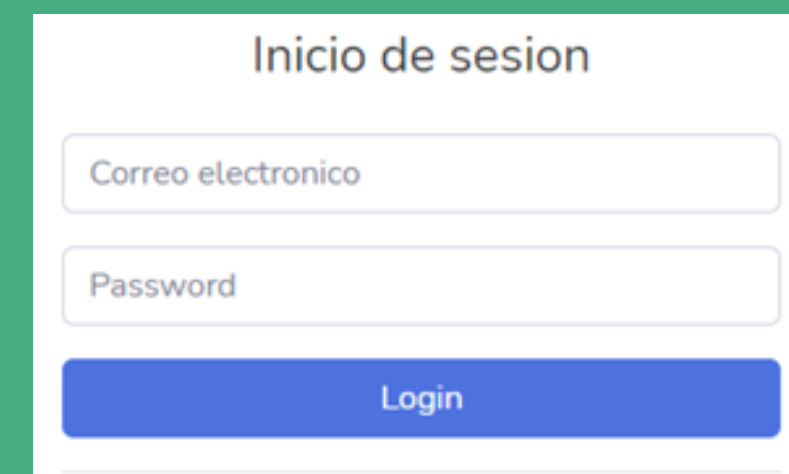
Metodología

-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.

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

```
<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>
```



Inicio de sesion

Correo electronico

Password

Login

Metodología

-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.

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

```
<?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();
```

```
<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>
```

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

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

Metodología

-Registrar usuario controller

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

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.

```
<?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'];

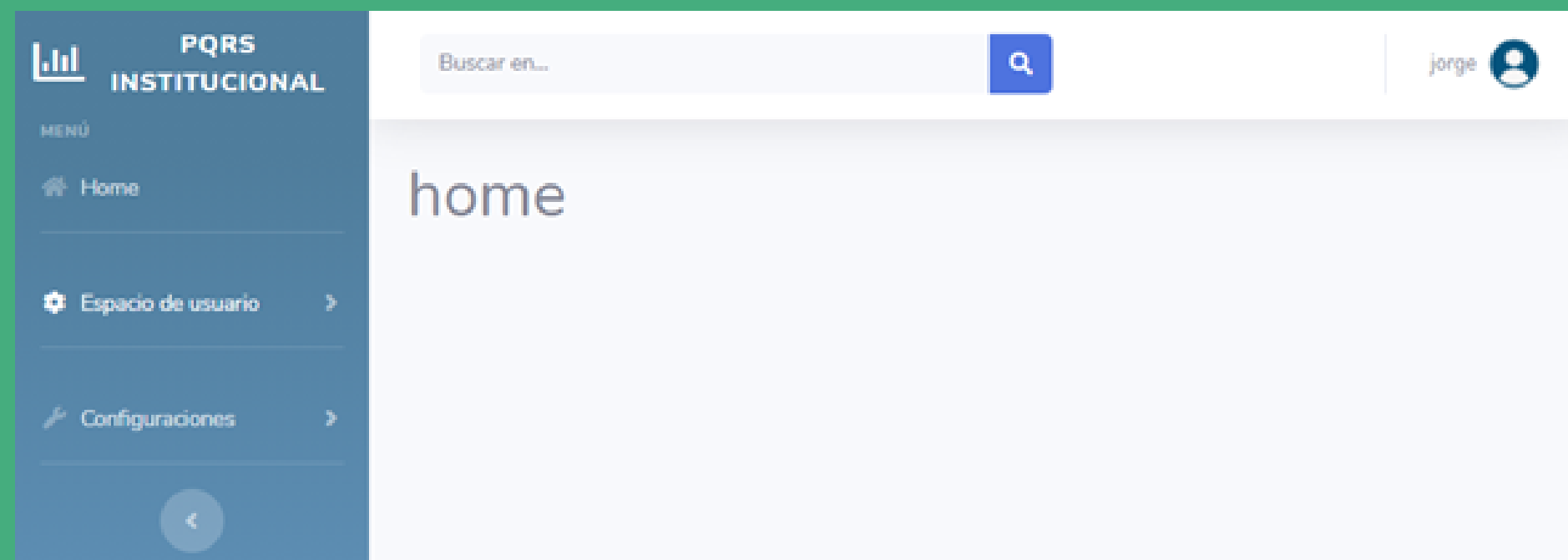
$response = array();
```

Metodología

-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



```
<> menu.1.view.html
<> menu.2.view.html
<> menu.3.view.html
```


Metodología

-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
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;
}
```

Metodología

-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
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;
}
```

Metodología

-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

De esta forma es como se validan los datos

```
<?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'];
```

```
$conexion->beginTransaction();

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

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.

Trabajo en equipo

- “Trabajar en equipo divide el trabajo y multiplica los resultados”
- “No preguntes qué puede hacer por ti el equipo. Pregunta qué puedes hacer tú por él”
- “Los logros de una organización son los resultados del esfuerzo combinado de cada individuo”