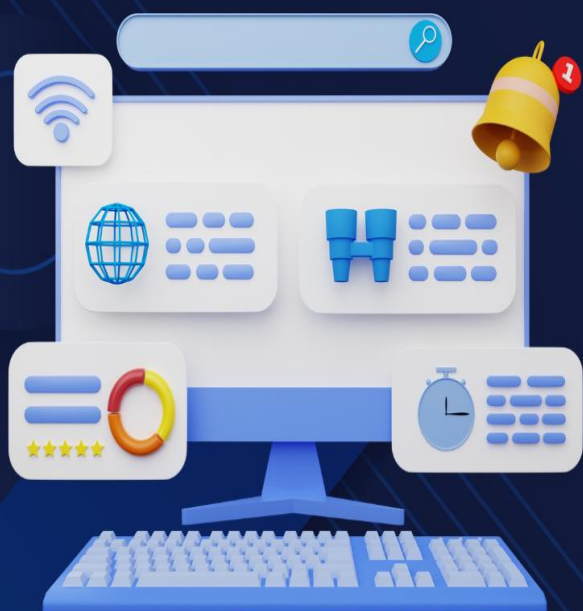




MANUAL TÉCNICO

Autor: Jonathan Andrei
Rodriguez Castillo

Tutor: Ing. Miriam Adriana
Castillo Merino



Migraciones necesarias para la creación de la base de datos.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('apellidos');
            $table->string('cedula')->unique();
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');

            $table->rememberToken();
            $table->timestamps();
        });

        Schema::create('password_reset_tokens', function (Blueprint $table) {
            $table->string('email')->primary();
            $table->string('token');
            $table->timestamp('created_at')->nullable();
        });

        Schema::create('sessions', function (Blueprint $table) {
            $table->string('id')->primary();
            $table->foreignId('user_id')->nullable()->index();
            $table->string('ip_address', 45)->nullable();
            $table->text('user_agent')->nullable();
            $table->longText('payload');
            $table->integer('last_activity')->index();
        });
    }

    /**
```

```

        * Reverse the migrations.
        */
    public function down(): void
    {
        Schema::dropIfExists('users');
        Schema::dropIfExists('password_reset_tokens');
        Schema::dropIfExists('sessions');
    }
};

```

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('cache', function (Blueprint $table) {
            $table->string('key')->primary();
            $table->mediumText('value');
            $table->integer('expiration');
        });

        Schema::create('cache_locks', function (Blueprint $table) {
            $table->string('key')->primary();
            $table->string('owner');
            $table->integer('expiration');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('cache');
        Schema::dropIfExists('cache_locks');
    }
};

```

```
}  
};
```

```
<?php  
  
use Illuminate\Database\Migrations\Migration;  
use Illuminate\Database\Schema\Blueprint;  
use Illuminate\Support\Facades\Schema;  
  
return new class extends Migration  
{  
    /**  
     * Run the migrations.  
     */  
    public function up(): void  
    {  
        Schema::create('jobs', function (Blueprint $table) {  
            $table->id();  
            $table->string('queue')->index();  
            $table->longText('payload');  
            $table->unsignedTinyInteger('attempts');  
            $table->unsignedInteger('reserved_at')->nullable();  
            $table->unsignedInteger('available_at');  
            $table->unsignedInteger('created_at');  
        });  
  
        Schema::create('job_batches', function (Blueprint $table) {  
            $table->string('id')->primary();  
            $table->string('name');  
            $table->integer('total_jobs');  
            $table->integer('pending_jobs');  
            $table->integer('failed_jobs');  
            $table->longText('failed_job_ids');  
            $table->mediumText('options')->nullable();  
            $table->integer('cancelled_at')->nullable();  
            $table->integer('created_at');  
            $table->integer('finished_at')->nullable();  
        });  
  
        Schema::create('failed_jobs', function (Blueprint $table) {  
            $table->id();  
            $table->string('uuid')->unique();  
            $table->text('connection');
```

```

        $table->text('queue');
        $table->longText('payload');
        $table->longText('exception');
        $table->timestamp('failed_at')->useCurrent();
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('jobs');
    Schema::dropIfExists('job_batches');
    Schema::dropIfExists('failed_jobs');
}
};

```

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('categories', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('slug')->unique();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {

```

```
        Schema::dropIfExists('categories');
    }
};
```

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('posts', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('slug')->unique();
            $table->text('extract')->nullable();
            $table->longText('body')->nullable();
            $table->enum('status', [1,2])->default(1);

            $table->unsignedBigInteger('user_id');
            $table->unsignedBigInteger('category_id');

            $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
            $table->foreign('category_id')->references('id')->on('categories')->onDelete('cascade');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('post');
    }
};
```

```
};
```

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('image', function (Blueprint $table) {
            $table->id();
            $table->string('url');
            $table->unsignedBigInteger('imagiable_id');
            $table->string('imagiable_type');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('image');
    }
};
```

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
```

```

/**
 * Run the migrations.
 */
public function up(): void
{
    $teams = config('permission.teams');
    $tableNames = config('permission.table_names');
    $columnNames = config('permission.column_names');
    $pivotRole = $columnNames['role_pivot_key'] ?? 'role_id';
    $pivotPermission = $columnNames['permission_pivot_key'] ??
'permission_id';

    if (empty($tableNames)) {
        throw new \Exception('Error: config/permission.php not loaded. Run
[php artisan config:clear] and try again.');
```

```

    }
    if ($teams && empty($columnNames['team_foreign_key'] ?? null)) {
        throw new \Exception('Error: team_foreign_key on
config/permission.php not loaded. Run [php artisan config:clear] and try
again.');
```

```

    }

    Schema::create($tableNames['permissions'], static function (Blueprint
$table) {
        // $table->engine('InnoDB');
        $table->bigIncrements('id'); // permission id
        $table->string('name');
        $table->string('description'); // For MyISAM use string('name',
225); // (or 166 for InnoDB with Redundant/Compact row format)
        $table->string('guard_name'); // For MyISAM use
string('guard_name', 25);
        $table->timestamps();

        $table->unique(['name', 'guard_name']);
    });

    Schema::create($tableNames['roles'], static function (Blueprint
$table) use ($teams, $columnNames) {
        // $table->engine('InnoDB');
        $table->bigIncrements('id'); // role id
        if ($teams || config('permission.testing')) { //
permission.testing is a fix for sqlite testing
            $table->unsignedBigInteger($columnNames['team_foreign_key'])-
>nullable();
            $table->index($columnNames['team_foreign_key'],
```

```

        Schema::drop($tableNames['roles']);
        Schema::drop($tableNames['permissions']);
    }
};

```

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('slides', function (Blueprint $table) {
            $table->id();
            $table->string('title', 100);
            $table->string('link')->nullable();
            $table->unsignedInteger('position')->default(0);

            $table->unsignedBigInteger('user_id');

            $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('slides');
    }
};

```

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('documents', function (Blueprint $table) {
            $table->id();
            $table->string('group_title');
            $table->string('title'); // título del documento
            $table->string('file_path');
            $table->json('variables')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('documents');
    }
};

```

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void

```

```

{
    Schema::create('document_submissions', function (Blueprint $table) {
        $table->id();

        $table->foreignId('document_id')->constrained();
        $table->foreignId('user_id')->constrained();

        $table->json('values'); // {"nombres":"Ana", ...}
        $table->string('output_path')->nullable();
        $table->string('pdf_path')->nullable();

        $table->enum('status', [
            'GENERADO', // Word generado y descargado
            'ACEPTADO',
            'RECHAZADO',
            'IMPRESO',
            'ENTREGADO',
        ]->default('GENERADO'));

        $table->timestamps();
    });
}
/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('document_submissions');
}
};

```

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void

```

```

{
    Schema::create('resources', function (Blueprint $table) {
        $table->id();

        $table->string('name'); // Nombre del recurso
        $table->string('code')->unique(); // Código institucional
        $table->string('category'); // Hardware, Software,
        Mobiliario, etc.
        $table->string('location')->nullable(); // Aula, laboratorio,
        etc.
        $table->string('status')->default('Disponible'); // Disponible,
        Prestado, Dañado
        $table->integer('quantity')->default(1); // Cantidad del recurso
        $table->string('assigned_to')->nullable(); // Nombre de quien lo
        tiene asignado
        $table->text('description')->nullable(); // Detalles técnicos u
        observaciones
        $table->foreignId('user_id')->constrained()->onDelete('cascade');

        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('resources');
}
};

```

Modelos

```

<?php

namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Category extends Model
{

```

```

use HasFactory;
protected $fillable = ['name','slug'];

public function getRouteKeyName()
{
    return 'slug';
}

public function posts(){
    return $this->hasMany(Post::class);
}
//Mostra la última publicación de la categoría, con estado 2 (publicada)
public function latestPost()
{
    return $this->hasOne(Post::class)
        ->where('status', 2)
        ->latest()
        ->with('image'); // Carga la imagen relacionada
}
}

```

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use App\Models\DocumentSubmission;
class Document extends Model
{
    public $timestamps = false;
    protected $fillable = ['group_title', 'title', 'file_path', 'variables'];

    protected $casts = [
        'variables' => 'array',
    ];
    public function submissions() { return $this->hasMany(DocumentSubmission::class); }
}

```

```

<?php

```

```

namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class DocumentSubmission extends Model
{
    protected $fillable = [
        'document_id',
        'user_id',
        'values',
        'output_path',
        'pdf_path',
        'status',
    ];

    protected $casts = [
        'values' => 'array',
    ];

    /* Relaciones */
    public function document()
    {
        return $this->belongsTo(Document::class);
    }

    public function user()
    {
        return $this->belongsTo(User::class);
    }
}

```

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Image extends Model
{
    use HasFactory;
    protected $fillable = ['url'];
}

```

```

        protected $table = 'image';
        //relacion polimorfica
        public function imagiable(){
            return $this->morphTo();
        }
    }
}

```

```

<?php

namespace App\Models;
use App\Models\User;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Post extends Model
{
    use HasFactory;
    protected $guarded = ['id', 'created_at', 'updated_at'];
    //
    public function user(){
        return $this->belongsTo(User::class);
    }
    public function category(){
        return $this->belongsTo(Category::class);
    }
    public function image(){
        return $this->morphOne(Image::class, 'imagiable');
    }
}

```

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Resource extends Model

```

```

{
    use HasFactory;

    protected $fillable = [
        'name', 'code', 'category', 'location', 'status',
        'quantity', 'assigned_to', 'description', 'user_id'
    ];

    public function user()
    {
        return $this->belongsTo(User::class);
    }
}

```

<?php

```

namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Slide extends Model
{
    use HasFactory;

    protected $guarded = ['id', 'created_at', 'updated_at'];
    public function user(){
        return $this->belongsTo(User::class);
    }

    public function image(){
        return $this->morphOne(Image::class, 'imagiable');
    }
}

```

```

    'apellidos',
    'cedula',

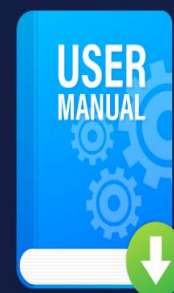
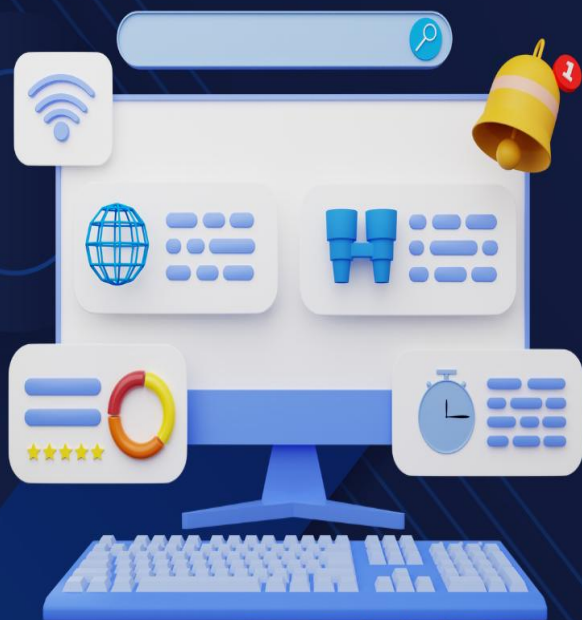
```



MANUAL DE USUARIO

Autor: Jonathan Andrei
Rodriguez Castillo

Tutor: Ing. Miriam Adriana
Castillo Merino



ÍNDICE

| | |
|---|----|
| OPCIONES GENERALES | 22 |
| INICIAR SESIÓN | 22 |
| REGISTRO DE NUEVO USUARIO | 22 |
| RESTABLECER CONTRASEÑA | 25 |
| VISUALIZAR DATOS DE PERFIL | 25 |
| ADMINISTRADOR..... | 28 |
| PANEL DE ADMISTRACIÓN | 28 |
| PAPEL DE ADMINISTRACION – LISTA DE USURIOS Y ASIGNACION DE ROLES | 29 |
| CREAR CATEGORÍA DE PUBLICACIÓN | 30 |
| VER PUBLICACIONES | 32 |
| CREAR PUBLICACIONES | 33 |
| ACTUALIZAR O EDITAR PUBLICACIONES | 34 |
| GESTIÓN DE DOCUMENTOS | 34 |
| SUBIR PLANTILLA | 34 |
| LISTA DE DOCUMENTOS | 36 |
| MALLA CURRICULAR | 37 |
| RECURSOS | 37 |
| CREAR RECURSO | 38 |
| DOCENTE. | 40 |
| CATEGORÍAS | 40 |
| CREAR CATEGORÍAS | 41 |
| LISTAR PUBLICACIONES | 41 |
| CREAR PUBLICACIONES | 42 |
| | 43 |

| | |
|---|----|
| GESTIÓN DE DOCUMENTOS | 43 |
| SUBIR PLANTILLA | 43 |
|
 | |
| Figura 1. Iniciar sesión | 22 |
| Figura 2. Registro de nuevo usuario | 23 |
| Figura 3. Alertas del formulario de registro | 24 |
| Figura 4. Envío de correo de verificación | 24 |
| Figura 5. Correo con link de verificación | 24 |
| Figura 6. Flujo para restablecer contraseña | 25 |
| Figura 7. Información de perfil y cambio de clave | 26 |
| Figura 8. Panel de administrador | 28 |
| Figura 9. Listado de usuarios | 29 |
| Figura 10. Asignar rol a usuario | 30 |
| Figura 11. Listado de categorías | 31 |
| Figura 12. Crear categorías | 31 |
| Figura 13. Formulario para la creación de esliker del carrusel de imágenes | 32 |
| Figura 14. Listado de publicaciones y acciones | 33 |
| Figura 15. Formulario para la creación de publicación | 33 |
| Figura 16. Actualizar publicación (formulario) | 34 |
| Figura 17. Ejemplo de plantillas a subir | 35 |
| Figura 18. Subir plantilla | 36 |
| Figura 19. Listar documentos | 36 |
| Figura 20. actualizar malla curricular | 37 |
| Figura 21. Vista de recursos | 38 |
| Figura 22. Ejemplo para crear recursos | 38 |
| Figura 23. Panel de administración rol docente | 40 |
| Figura 24. Listar categorías | 41 |
| Figura 25. crear categoría rol docente | 41 |
| Figura 26. Listado de publicaciones | 42 |
| Figura 27. Formulario para la creación de publicación rol docente | 43 |

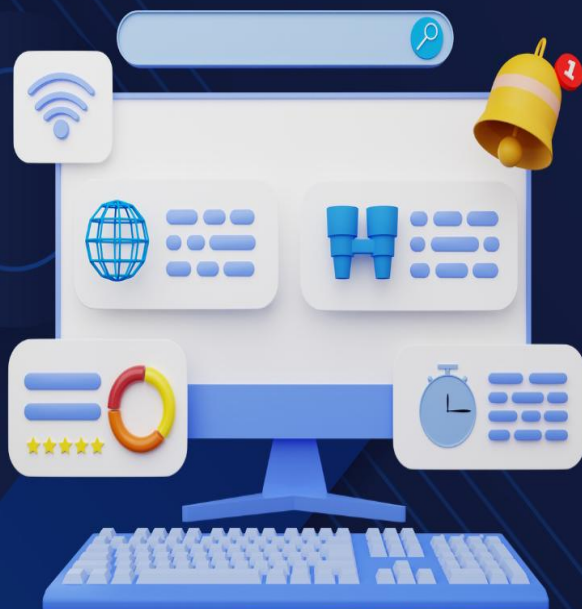
| | |
|---|----|
| Figura 28. Ejemplo de subir plantilla al sistema | 43 |
| Figura 29. Panel de administración rol estudiante | 45 |
| Figura 30. Formulario para solicitud de documentos estudiantes | 45 |



OPCIONES GENERALES

Autor: Jonathan Andrei
Rodriguez Castillo

Tutor: Ing. Miriam Adriana
Castillo Merino



OPCIONES GENERALES

INICIAR SESIÓN

Para acceder al Sistema de gestión se debe de ingresar al URL <https://unesum-tic.es/login> , se mostrará el formulario para iniciar sesión, donde se deberá de llenar los campos correo y contraseña, como se muestra en la siguiente **Figura 1**

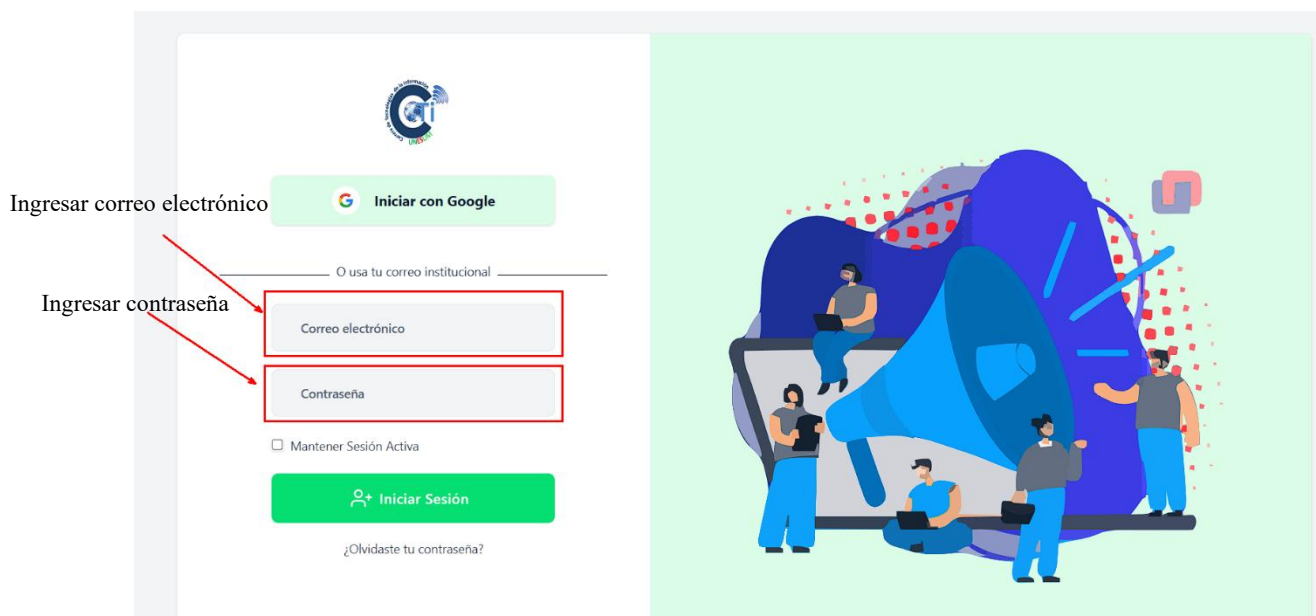


Figura 1. Iniciar sesión

Nota: Las personas autenticadas y verificadas podrán acceder al sistema caso contrario deberán registrarse.

REGISTRO DE NUEVO USUARIO

El usuario deberá dirigirse a la barra de navegación opción registrase donde deberá proporcionar sus datos como nombres, apellidos, numero de cédula, contraseña y confirmación de contraseña, una vez con estos datos llenos en el formulario se registrara, sin embargo aun no podrá acceder ninguna funcionalidad sin que se haya verificado su dirección de correo electrónico es por ello que el sistema enviara un correo al buzón del correo registrado por el usuario, contendrá un link con el cual puede verificar su cuenta.

Crear una cuenta

NOMBRES

Nombres

APELLIDOS

Apellidos

NUMERO DE CEDULA

Cédula

CORREO AL QUE SE ENVIA LINK DE VERIFICACION

Correo

ESCRIBIR CONTRASEÑA

Contraseña

REESCRIBIR CONTRASEÑA PARA CONFIRMAR ELECCION

Confirmar contraseña

¿Ya tienes cuenta?

REGISTRARSE

Figura 2. Registro de nuevo usuario

El sistema cifra la contraseña ingresada y valida que los campos estén rellenos, si el campo cedula esta mal escrito el sistema alertara con un mensaje, de igual forma para el campo contraseña no debe tener menos de 8 caracteres y ser alfanumérico.

Crear una cuenta

Nombres
Jonathan

Apellidos
Castillo

Cédula
0956523311
La cédula no es válida.

Correo
boris123@hotmail.com

Contraseña
La confirmación de password no coincide.

Confirmar contraseña

¿Ya tienes cuenta?

REGISTRARSE

Verificación de correo electrónico

¡Gracias por registrarte! Antes de comenzar, por favor verifica tu dirección de correo haciendo clic en el enlace que te hemos enviado. Si no recibiste el correo, con gusto te enviaremos otro.

[REENVIAR CORREO DE VERIFICACIÓN](#)[Cerrar sesión](#)

Figura 5. Correo con link de verificación

RESTABLECER CONTRASEÑA

El caso de que el usuario haya olvidado su contraseña, deberá restablecerla en el formulario de inicio de sesión opciones ¿Olvidaste tu contraseña?, el usuario proporcionarla su dirección de correo electrónico en la cual se enviara el link que lo redirigirá al formulario de reseteo de contraseña.

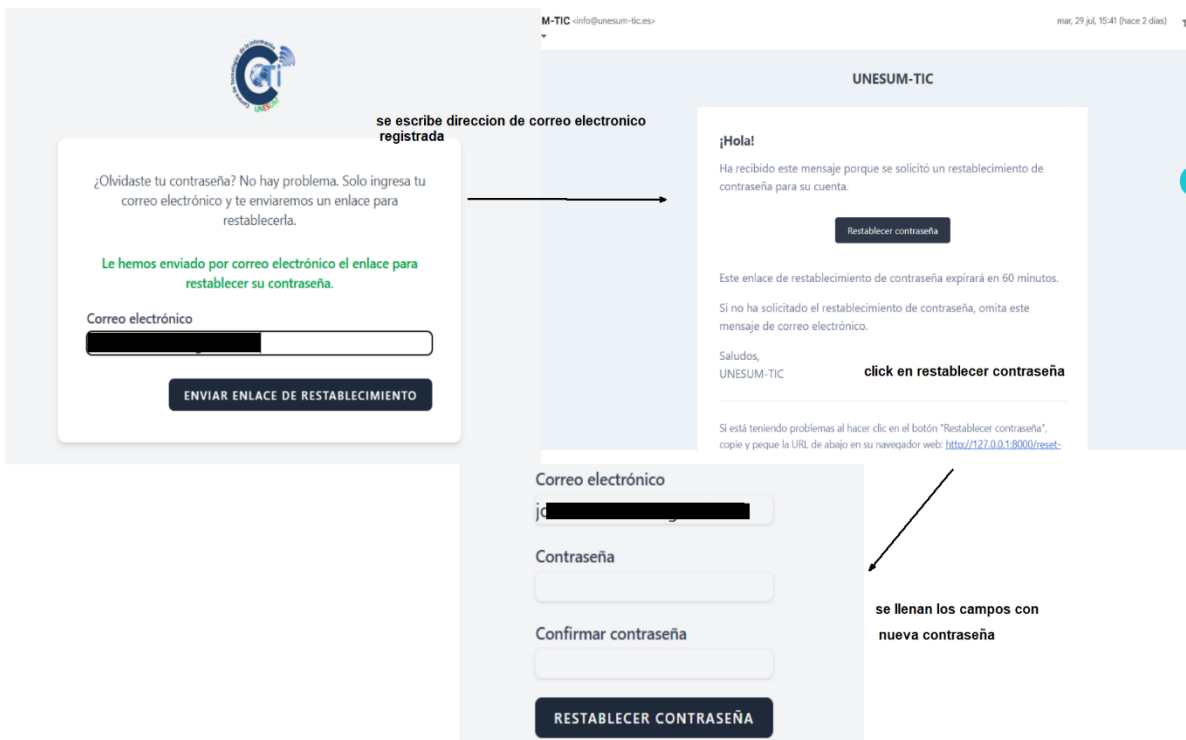


Figura 6. Flujo para restablecer contraseña

VISUALIZAR DATOS DE PERFIL.

Con los pasos anteriores el usuario podrá acceder a la información de su perfil que tendrá datos como nombres, correo y la opción de cambiar contraseña.

Mi perfil

Panel de administración

configuraciones

Cerrar sesión

Información de perfil

Actualice la información de su cuenta y la dirección de correo electrónico.

Nombre

Correo electrónico

GUARDAR

Actualizar contraseña

Asegúrese que su cuenta esté usando una contraseña larga y aleatoria para mantenerse seguro.

Contraseña actual

Nueva Contraseña

Confirmar contraseña

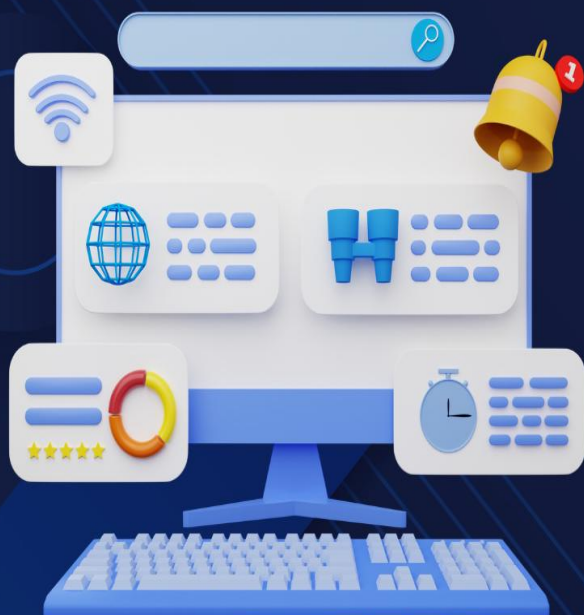
GUARDAR

Figura 7. Información de perfil y cambio de clave

ADMINISTRADOR

Autor: Jonathan Andrei
Rodriguez Castillo

Tutor: Ing. Miriam Adriana
Castillo Merino



ADMINISTRADOR

PANEL DE ADMISTRACIÓN.

Una vez el usuario administrador se haya autenticado y su cuenta verificada, podrá acceder al panel de administración donde se muestran como interfaz principal el dashboard detallando estadísticas como porcentaje de usuarios registrado por roles, también las demás funcionalidades como publicaciones, datos de usuarios como nombres y correos registrados, listado de roles, slider de la página principal, gestión de documentos para solicitar, subir plantilla, gestionar inventario.

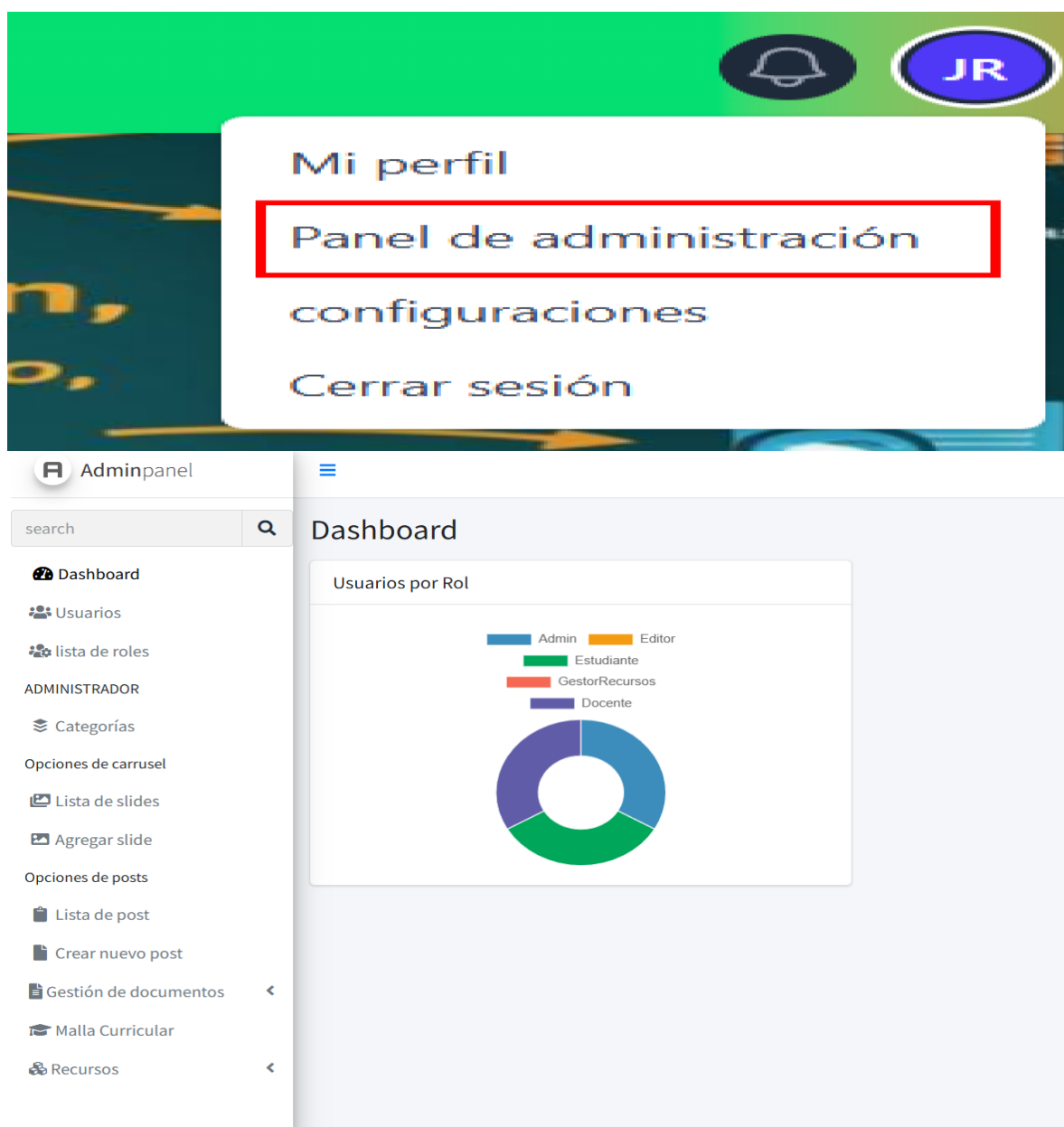
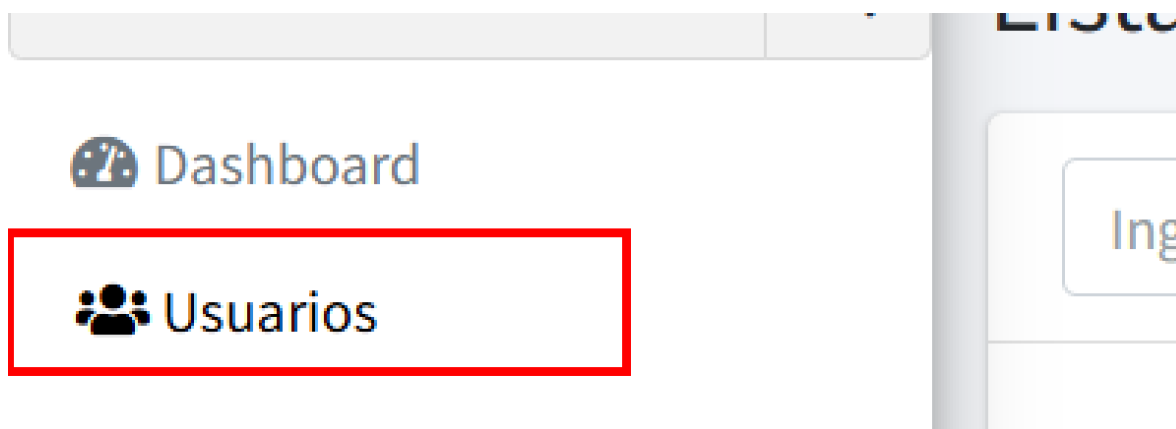


Figura 8. Panel de administrador

PAPEL DE ADMINISTRACION – LISTA DE USURIOS Y ASIGNACION DE ROLES.

En esta sección se podrá ver los usuarios registrados en el sistema, se puede editar el rol de usuario puede tener uno o más roles.



Click en la acción editar para asignar rol al usuario seleccionado

Lista de usuarios

Ingrese el nombre o correo del usuario

| ID | Nombre | Email | Acciones |
|----|--------------------|-------------------------------------|------------------------|
| 1 | Jonathan Rodriguez | jonathanbz500@gmail.com | Editar |
| 2 | Sully Calderon | sullycalderon20@gmail.com | Editar |
| 3 | Oscar Acuña | acuna-oscar2365@unesum.edu.ec | Editar |
| 4 | Alexander Cabezas | cabezas-alexander0103@unesum.edu.ec | Editar |

Figura 9. Listado de usuarios

Escoger el rol correspondiente, se muestra el listado, click en asignar rol

Asignar un rol

Nombre:

Oscar Acuña

☐ **Admin**

☐ **Editor**

☐ **Estudiante**

☐ **GestorRecursos**

Asignar rol

Figura 10. Asignar rol a usuario

CREAR CATEGORÍA DE PUBLICACIÓN

Antes de la publicación de un comunicado se deberá agregar alguna categoría perteneciente evento; festividades, casas abiertas, entre otras.

En el apartado de categorías se pueden apreciar las ya creadas y de necesitar agregar alguna

ADMINISTRADOR

 **Categorías**

En esta sección se muestran las categorías ya creadas, con opciones a editar, es decir, cambiar el nombre de la misma, a su vez también se muestra la opción de agregar categoría

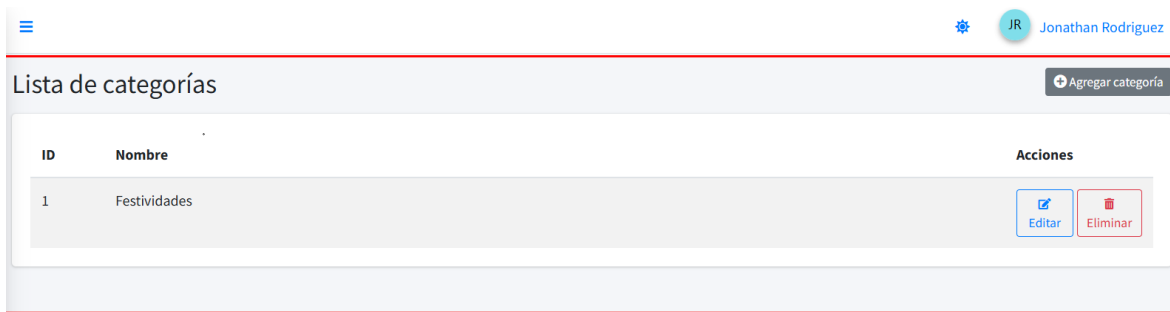


Figura 11. Listado de categorías

Al dar click en agregar categoría se mostrará el formulario, el cual solo tenga un campo que llenar: nombre de categoría el slug se rellena automáticamente, este se usa para el URL de la publicación.

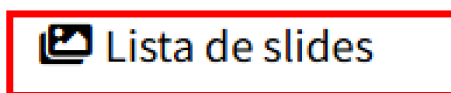
A screenshot of a web application interface titled "Crear categoría". It contains two input fields. The first field is labeled "Nombre" and has a placeholder text "Ingrese el nombre". The second field is labeled "Slug" and has a placeholder text "Ingrese el slug". Below these fields is a blue button with the text "Añadir Categoría".

Figura 12. Crear categorías

SLIDERS

En esta sección el usuario podrá ubicar los diferentes slider o imágenes que se muestran en el carrusel en la página principal, se tiene un limite de 12 slider después de esto solo se podrán editar o eliminar los ya existentes si se requiere cambiar o agregar más sliders.

Opciones de carrusel



 Agregar slide

| Lista de Slides | | | | |
|--|---|--|---------|--|
| <div> <div>Start</div> <div>Crear Slide</div> </div> | | | | |
| ID | Título | Imagen | URL | Acciones |
| 2 | Bienvenidos a la carrera de Tecnologías de la Información |  | Sin URL | <div> <div>Editar</div> <div>Eliminar</div> </div> |

Crear nuevo Slide

Start

Título *

Anuncio

Enlace (opcional)

Previsualización de Imagen



Imagen *

Seleccionar archivo

WhatsApp Image 2025-07-31 at 1.05.25 AM (1).jpeg

Guardar Slide

Cancelar

Figura 13. Formulario para la creación de eslider del carrusel de imágenes

La imagen del carrusel podrá o no tener enlace que redirija alguna publicación realizada en el mismo sistema o externo a otras paginas web

VER PUBLICACIONES

Deberá dirigirse a la opción de Lista de post, donde al dar click aparecerán los posts creados por el usuario, tendrá las acciones de aditar o eliminar publicación

Opciones de posts



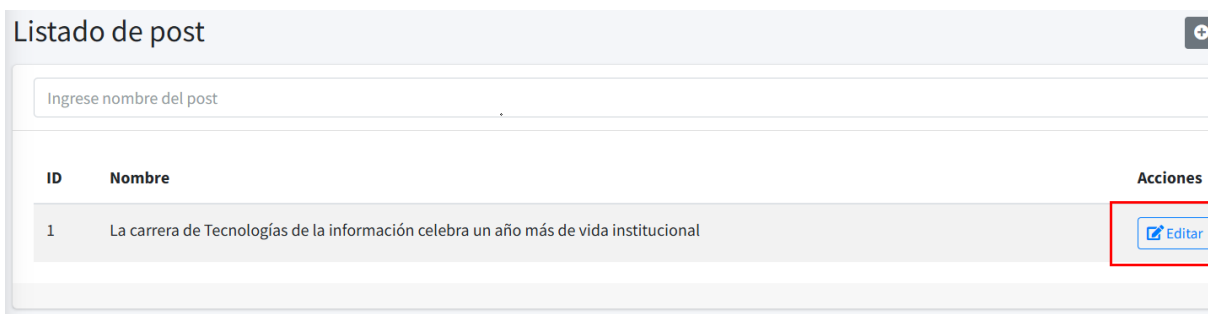
Lista de post



Crear nuevo post

ACTUALIZAR O EDITAR PUBLICACIONES

En esta sección el usuario podrá actualizar el contenido de la publicación en el caso que requiera corregir algún error o actualizar la información del mismo.



Aparecerá el mismo formulario que cuando se creó la publicación simplemente reemplace la información que desee cambiar y dar click en actualizar

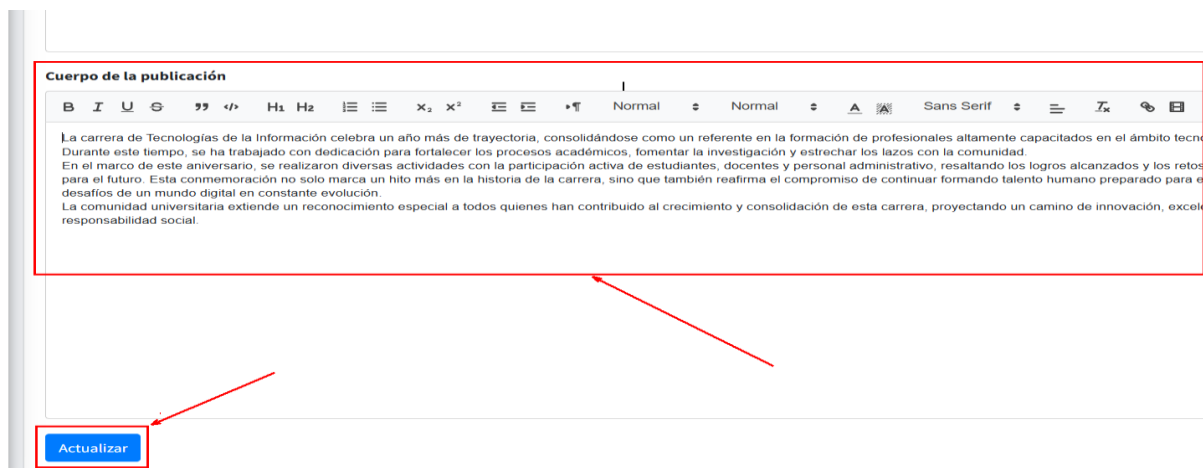


Figura 16. Actualizar publicación (formulario)

GESTIÓN DE DOCUMENTOS

SUBIR PLANTILLA

En esta sección se puede subir la plantilla Word de documentos a rellenar de forma automática por estudiantes, en las secciones del documento que deberían variar o rellenarse se

deberán ubicar variables con el siguiente formato para que el programa pueda extraerlas y crear formularios dinámicos en base a las mismas.

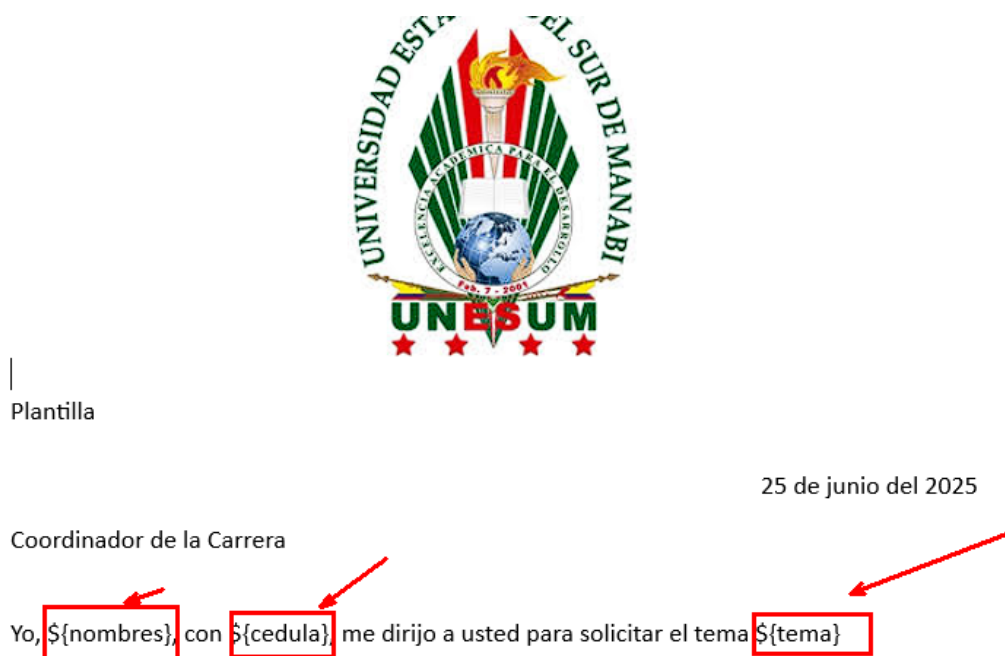


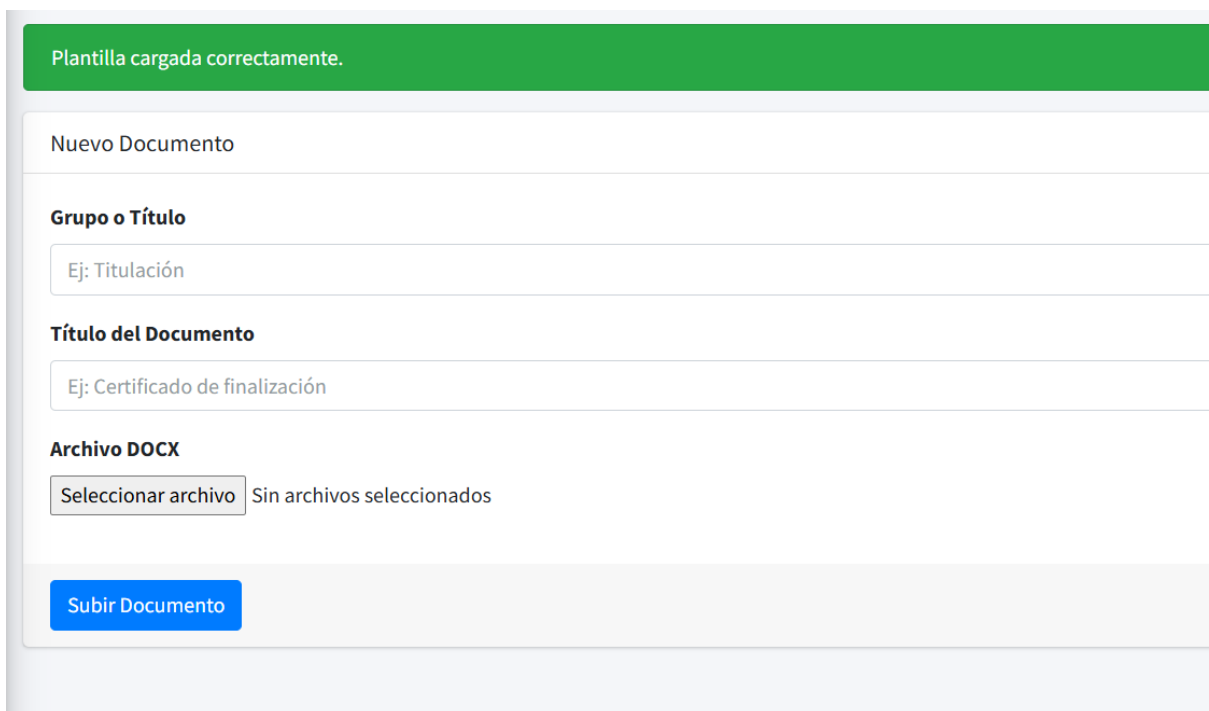
Figura 17. Ejemplo de plantillas a subir

Gestión de documentos

 Subir Documento

 Lista de Documentos

El formulario para subir plantillas conta de 3 campos a llenar el grupo al que pertenece el documento por ejemplo titulación, vinculación, entre otros,



Plantilla cargada correctamente.

Nuevo Documento

Grupo o Título

Ej: Titulación

Título del Documento

Ej: Certificado de finalización

Archivo DOCX

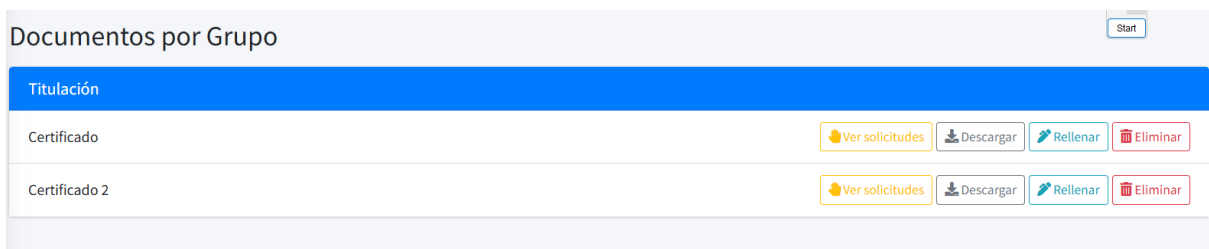
Seleccionar archivo Sin archivos seleccionados

Subir Documento

Figura 18. Subir plantilla

LISTA DE DOCUMENTOS

Esta función permite listar los documentos subidos y poder realizar opciones como descargar, rellenar o eliminar los documentos.



| Documentos por Grupo | | | | | Start |
|----------------------|--|-----------------|-----------|----------|----------|
| Titulación | | | | | |
| Certificado | | Ver solicitudes | Descargar | Rellenar | Eliminar |
| Certificado 2 | | Ver solicitudes | Descargar | Rellenar | Eliminar |

Figura 19. Listar documentos

MALLA CURRICULAR.

Sí el usuario requiere de actualizar el PDF con la malla curricular por cambios en la misma podrá realizarlo, sin embargo será necesario que el nuevo documento tenga el mismo nombre que el anterior para que el sistema lo actualice con éxito.

Editar Malla Curricular

Archivo PDF de la Malla

Seleccionar archivo Sin archivos seleccionados

Actualizar

Vista previa del documento actual:

MALLA-CURRICULAR-CTI-8-NIVELES.pdf 1 / 13 100%



Universidad Estatal del Sur de Manabí
Creada el 7 de Febrero del año 2001, según Registro Oficial No. 261
CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

MALLA CURRICULAR - CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Facultad Ciencias Técnicas
Carrera: Tecnologías de la Información

Distribución de Asignaturas por Unidades de Organización Curricular

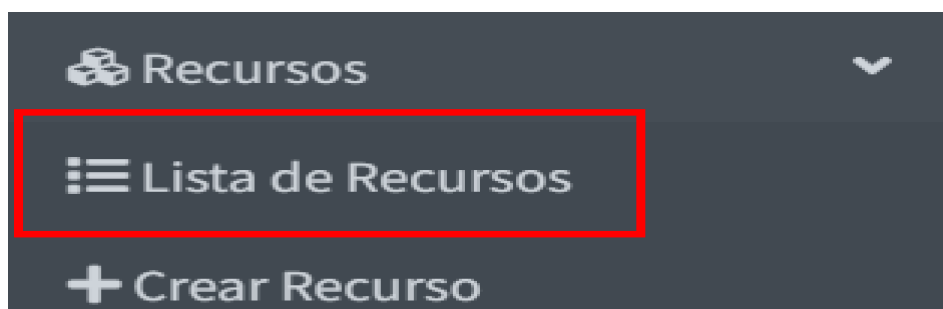
| Nivel | Unidad | Asignatura | Créditos | Horas | Prerrequisitos |
|-------|--------|-------------------------------|----------|-------|----------------|
| I | I | Matemática I | 3 | 90 | |
| | | Introducción a la Informática | 3 | 90 | |
| II | II | Matemática II | 3 | 90 | |
| | | Física | 3 | 90 | |
| III | III | Investigación | 3 | 90 | |
| | | Programación I | 3 | 90 | |
| IV | IV | Algoritmos y Lenguajes | 3 | 90 | |
| | | Programación II | 3 | 90 | |

Figura 20. actualizar malla curricular

RECURSOS

LISTA DE RECURSOS

Esta sección permite visualizar los recursos registrados y filtrar los mismos según el nombre o código del mismo



| Lista de Recursos | | | | | | | |
|---|--------|-----------|------------|----------|---------------|--------------------|--|
| <div> <div>✓ Éxito</div> <div>Recurso creado correctamente</div> </div> | | | | | | | |
| <div> <div>Buscar por nombre o código</div> </div> | | | | | | | |
| Nombre | Código | Categoría | Estado | Cantidad | Ubicación | Asignado a | Acciones |
| Pc_Escritorio | 777d7 | Equipo | Disponible | 1 | Laboratorio 1 | Jonathan Rodriguez | <div> <div>✎ Editar</div> <div>✖ Eliminar</div> </div> |

Figura 21. Vista de recursos

CREAR RECURSO.

En esta sección el administrador podrá agregar al inventario un elemento, llenando los campos como se muestra en la siguiente figura:

| | |
|--------------------|--------------------|
| Nombre | Laptop |
| Código | 748599 |
| Categoría | Equipo tecnológico |
| Ubicación | Laboratorio 1 |
| Estado | Disponible |
| Cantidad | 1 |
| Asignado a | Boris Rodriguez |
| Descripción | |
| Guardar | Cancelar |

Figura 22. Ejemplo para crear recursos

DOCENTE

Autor: Jonathan Andrei
Rodriguez Castillo

Tutor: Ing. Miriam Adriana
Castillo Merino



DOCENTE.

PANEL DE ADMINISTRACIÓN.

Una vez el usuario docente se haya autenticado y su cuenta verificada, podrá acceder al panel de administración donde se muestran como interfaz principal el dashboard detallando estadísticas como porcentaje de usuarios registrado por roles, también las demás funcionalidades como publicaciones, gestión de documentos.

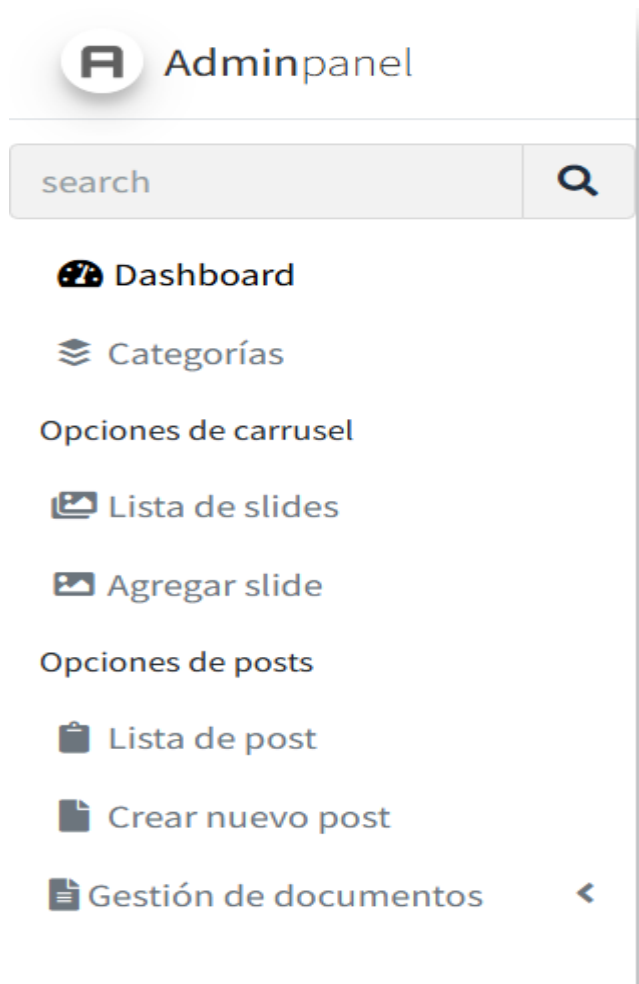
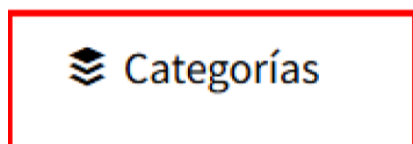


Figura 23. Panel de administración rol docente

CATEGORÍAS

Al igual que el usuario administrador el usuario docente podrá listar y crear diferentes categorías para las publicaciones



Lista de categorías

+ Agregar categoría

| ID | Nombre | Acciones |
|----|--------------|---|
| 1 | Festividades | <div><div>Editar</div><div>Eliminar</div></div> |

CREAR CATEGORÍAS

Se rellena el único campo el nombre de la categoría

Crear categoría

Nombre

Slug

Añadir Categoría

Figura 25. crear categoría rol docente

PUBLICACIONES

Del mismo modo que con el usuario administrador una vez creada la categoría a la que pertenece se podrá escoger en el formulario de la publicación

LISTAR PUBLICACIONES

Deberá dirigirse a la opción de Lista de post, donde al dar click aparecerán los posts creados por el usuario, tendrá las acciones de aditar o eliminar publicación

Opciones de posts



Lista de post



Crear nuevo post

Listado de post

Nueva Publicación

Ingrese nombre del post

| ID | Nombre | Acciones |
|----|--|---|
| 1 | La carrera de Tecnologías de la información celebra un año más de vida institucional | <div><div>Editar</div><div>Eliminar</div></div> |

Figura 26. Listado de publicaciones

Cabe destacar que los usuarios en general solo podrán visualizar publicaciones creadas por el mismo

CREAR PUBLICACIONES

Es usuario al dar click en la opción de crear nuevo post se presentará el formulario para la creación del post, deberá rellenar los campos; nombre o título de la publicación, escoger la categoría a la que pertenece, el estado si es un borrador o ya se a de publicar, la imagen de post, el extracto que será un pequeño resumen o introducción del tema, cuerpo de la publicación texto completo de la publicación.

| | |
|---|--|
| Nombre | |
| La carrera de Tecnologías de la información celebra un año más de vida institucional | |
| Slug | |
| la-carrera-de-tecnologias-de-la-informacion-celebra-un-año-más-de-vida-institucional | |
| Categoría | |
| Facultades | |
| Estado | |
| ☐ Borrador ☒ Publicado | |
|  | <p>Cargar imagen que se mostrará en la publicación</p> <p>Seleccionar archivo Sin archivos seleccionados</p> <p>Indicaciones</p> |
| Extracto | |
| <p>Con orgullo, la carrera de Tecnologías de la información conmemora un nuevo aniversario de compromiso académico y crecimiento institucional.</p> | |

Extracto

Con orgullo, la carrera de Tecnologías de la Información conmemora un nuevo aniversario de compromiso académico y crecimiento institucional.

Cuerpo de la publicación

La carrera de Tecnologías de la Información celebra un año más de trayectoria, consolidándose como un referente en la formación de profesionales altamente capacitados en el ámbito tecnológico. Durante este tiempo, se ha trabajado con dedicación para fortalecer los procesos académicos, fomentar la investigación y estrechar los lazos con la comunidad. En el marco de este aniversario, se realizan diversas actividades con la participación activa de estudiantes, docentes y personal administrativo, resaltando los logros alcanzados y los retos asumidos para el futuro. Esta conmemoración no solo marca un hito más en la historia de la carrera, sino que también reafirma el compromiso de continuar formando talento humano preparado para enfrentar los desafíos de un mundo digital en constante evolución. La comunidad universitaria respalda el reconocimiento a todos quienes han contribuido al crecimiento y consolidación de esta carrera, proyectando un camino de innovación, excelencia y responsabilidad social.

[Publicar](#)

Figura 27. Formulario para la creación de publicación rol docente

GESTIÓN DE DOCUMENTOS

SUBIR PLANTILLA

En esta sección se puede subir la plantilla Word de documentos a rellenar de forma automática por estudiantes, en las secciones del documento que deberían variar o rellenarse se deberán ubicar variables con el siguiente formato para que el programa pueda extraerlas y crear formularios dinámicos en base a las mismas.



Plantilla

25 de junio del 2025

Coordinador de la Carrera

Yo, S{nombres}, con S{cedula} me dirijo a usted para solicitar el tema S{tema}

Figura 28. Ejemplo de subir plantilla al sistema

ESTUDIANTE

Autor: Jonathan Andrei
Rodriguez Castillo

Tutor: Ing. Miriam Adriana
Castillo Merino



ESTUDIANTE.

Una vez el usuario docente se haya autenticado y su cuenta verificada, podrá acceder al panel de administración donde se muestran como interfaz principal el panel de administración con la opción de listar documentos en esta el usuario podrá rellenar y enviar documentos que requiera de la carrera.

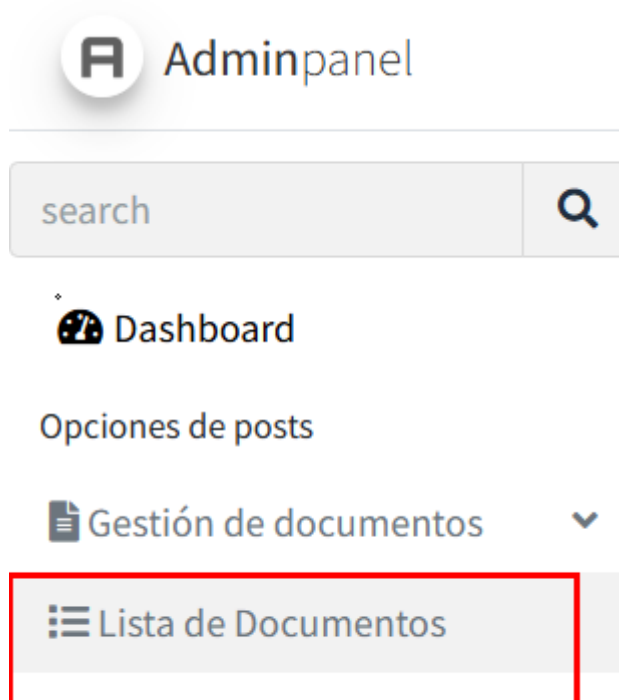


Figura 29. Panel de administración rol estudiante

RELLENAR FORMULARIO PARA SOLICITUD DE DOCUMENTO

Es esta sección de listar se mostrarán los documentos subidos al sistema por administradores o docentes.

The image shows a form titled 'Titulación » Certificado'. It contains three input fields labeled 'Nombres', 'Cedula', and 'Tema'. Below these fields is a blue button labeled 'Generar y descargar Word'.

Figura 30. Formulario para solicitud de documentos estudiantes

Una vez se genere el documento se descarga en formato docx (word) transformar en PDF, y subir en el formulario:

Subir PDF convertido

Selecciona tu archivo PDF

Seleccionar archivo

Sin archivos seleccionados

 Subir PDF