# Documentación del Proceso de Desarrollo

## Introducción

Este documento describe el proceso de desarrollo del sistema de registro de cursos. Incluye la estructura de la base de datos,   
el diseño de las páginas HTML, las funcionalidades implementadas en PHP y las validaciones realizadas para garantizar un sistema funcional y seguro.

## Estructura de la Base de Datos

La base de datos contiene las siguientes tablas principales:  
1. `Courses`: Almacena la información de los cursos disponibles, incluyendo el nombre, descripción, número total de plazas y plazas disponibles.  
2. `Enrollments`: Registra las inscripciones de los estudiantes a los cursos. Incluye información del estudiante y del curso al que se inscribe.

Script SQL utilizado:

CREATE DATABASE IF NOT EXISTS CourseRegistration;  
USE CourseRegistration;  
  
CREATE TABLE IF NOT EXISTS Courses (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 name VARCHAR(100) NOT NULL,  
 description TEXT,  
 total\_slots INT NOT NULL,  
 available\_slots INT NOT NULL,  
 image\_path VARCHAR(255)  
);  
  
CREATE TABLE IF NOT EXISTS Enrollments (  
 id INT AUTO\_INCREMENT PRIMARY KEY,  
 course\_id INT,  
 student\_name VARCHAR(100) NOT NULL,  
 email VARCHAR(100) NOT NULL,  
 cedula VARCHAR(10) NOT NULL,  
 edad INT NOT NULL,  
 genero VARCHAR(20) NOT NULL,  
 FOREIGN KEY (course\_id) REFERENCES Courses(id)  
);

## Diseño de las Páginas HTML

El sistema incluye las siguientes páginas principales:  
1. `index.php`: Página principal que lista los cursos disponibles, mostrando el número de plazas ocupadas y disponibles.  
2. `registro.php`: Página de registro donde los usuarios pueden inscribirse en los cursos.  
3. `consulta.php`: Página que permite consultar las inscripciones realizadas y gestionar la eliminación de registros si es necesario.

## Código PHP Utilizado

Se implementaron funcionalidades clave en PHP para conectar el frontend con la base de datos y manejar la lógica del sistema:  
1. Validación de datos del usuario, como la cédula y campos requeridos.  
2. Actualización de las plazas disponibles en la tabla `Courses` tras cada registro.  
3. Eliminación de inscripciones y restauración de plazas en caso de eliminar un registro.  
4. Mensajes flotantes para confirmar las acciones realizadas.

Ejemplo de un fragmento de código PHP para registro de usuarios:

<?php  
$conn = new mysqli("localhost", "root", "", "CourseRegistration");  
if ($conn->connect\_error) {  
 die("Error en la conexión: " . $conn->connect\_error);  
}  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
 $course\_id = $\_POST["curso\_id"];  
 $student\_name = $\_POST["student\_name"];  
 $email = $\_POST["email"];  
 $cedula = $\_POST["cedula"];  
 $edad = $\_POST["edad"];  
 $genero = $\_POST["genero"];  
  
 $check\_slots = $conn->query("SELECT available\_slots FROM Courses WHERE id = $course\_id");  
 $row = $check\_slots->fetch\_assoc();  
 if ($row["available\_slots"] > 0) {  
 $conn->query("INSERT INTO Enrollments (course\_id, student\_name, email, cedula, edad, genero) VALUES ('$course\_id', '$student\_name', '$email', '$cedula', '$edad', '$genero')");  
 $conn->query("UPDATE Courses SET available\_slots = available\_slots - 1 WHERE id = $course\_id");  
 } else {  
 echo "No hay plazas disponibles.";  
 }  
}  
?>

## Validaciones y Funcionalidades Adicionales

1. \*\*Validación de Cédula Ecuatoriana\*\*: Se implementó un algoritmo para validar que las cédulas ingresadas sean válidas según el formato nacional.  
2. \*\*Gestión de Plazas Dinámicas\*\*: Las plazas ocupadas y disponibles se actualizan automáticamente en la interfaz y la base de datos.  
3. \*\*Mensajes Flotantes\*\*: Se muestran mensajes flotantes para confirmar acciones como registros exitosos.

## Conclusión

El sistema desarrollado cumple con los requisitos propuestos y permite la gestión eficiente de cursos e inscripciones. Además, se implementaron validaciones y mejoras en la experiencia del usuario, garantizando un sistema robusto y funcional.