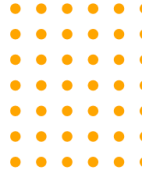


Universidad Politécnica del Estado de
Morelos



EP4. SITIO WEB DEL LADO DEL SERVIDOR

Docente: César Iván Ponciano Velasquez

Materia: Programación web

Alumnos:

*Flores Zarco Edna Jaqueline
Vásquez Estrada Juan Alberto*

7°C 21/11/2025



ÍNDICE

Introducción.....	3
Respaldo de la base de datos.....	4
Recuperación de la base de datos.....	5
Reportes en PDF.....	6
Reportes con gráficas en PDF.....	7
Pruebas.....	8
Conclusiones.....	9

Introducción

El presente documento técnico describe la implementación de una aplicación web para la gestión de competencias con funcionalidades avanzadas de respaldo, recuperación y generación de reportes. En el desarrollo de aplicaciones web modernas, es fundamental contar con mecanismos robustos que garanticen la integridad y disponibilidad de la información, así como herramientas que permitan la visualización y análisis de datos de manera eficiente.

Este sistema ha sido desarrollado utilizando el patrón de arquitectura Modelo-Vista-Controlador (MVC) en PHP, implementando tres funcionalidades principales: un módulo de respaldo automático de bases de datos, un sistema de recuperación de información desde archivos de respaldo, y un generador de reportes en formato PDF con capacidad de integrar gráficas estadísticas.

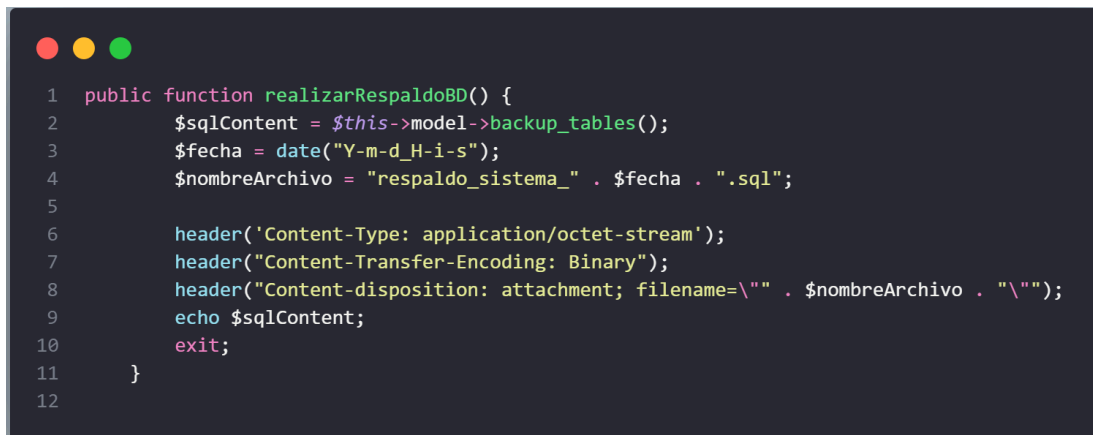
El objetivo de esta documentación es presentar de manera detallada la estructura del código, las pruebas realizadas y los resultados obtenidos, demostrando la efectividad de las soluciones implementadas para la gestión integral de datos en aplicaciones web empresariales.

Respaldo de la base de datos.

```
1  $tables = [];  
2  $result = $this->connection->query('SHOW TABLES');  
3  while($row = $result->fetch_row()) {  
4      $tables[] = $row[0];  
5  }  
6  
7  $return = "SET FOREIGN_KEY_CHECKS=0;\n\n";  
8  
9  foreach($tables as $table) {  
10     $result = $this->connection->query('SELECT * FROM ' . $table);  
11     $num_fields = $result->field_count;  
12  
13     $return .= "DROP TABLE IF EXISTS `". $table . "`;\n";  
14     $row2 = $this->connection->query('SHOW CREATE TABLE ' . $table)->fetch_row();  
15     $return .= $row2[1] . ";\n\n";  
16  
17     for ($i = 0; $i < $num_fields; $i++) {  
18         while($row = $result->fetch_row()) {  
19             $return .= 'INSERT INTO ' . $table . ' VALUES(';  
20             for($j = 0; $j < $num_fields; $j++) {  
21  
22                 if (isset($row[$j])) {  
23                     $row[$j] = addslashes($row[$j]);  
24                     $row[$j] = str_replace("\n", "\\n", $row[$j]);  
25                     $return .= "' . $row[$j] . '";  
26                 } else {  
27                     $return .= 'NULL';  
28                 }  
29  
30                 if ($j < ($num_fields - 1)) {  
31                     $return .= ',';  
32                 }  
33             }  
34             $return .= ");\n";  
35         }  
36     }  
37     $return .= "\n\n";  
38 }  
39  
40  
41  
42 $fecha = date("Y-m-d");  
43 $handle = fopen('config/backups/db-backup-' . $fecha . '.sql', 'w+');  
44 fwrite($handle, $return);  
45 fclose($handle);  
46  
47 $return .= "SET FOREIGN_KEY_CHECKS=1;";  
48 return $return;  
49 }
```

Img. 1 Código del respaldo en UserModel.php

En la imagen número 1 se muestra el código PHP del método de respaldo de base de datos implementado en la clase UserModel.php, que contiene la funcionalidad para exportar la estructura y los datos de la base de datos.



```

1  public function realizarRespaldoBD() {
2      $sqlContent = $this->model->backup_tables();
3      $fecha = date("Y-m-d_H-i-s");
4      $nombreArchivo = "respaldo_sistema_" . $fecha . ".sql";
5
6      header('Content-Type: application/octet-stream');
7      header("Content-Transfer-Encoding: Binary");
8      header("Content-disposition: attachment; filename=\"\" . $nombreArchivo . \"\");
9      echo $sqlContent;
10     exit;
11 }
12

```

Img. 2 Código del respaldo en UserController.php

En la imagen número 2 se muestra el código PHP del controlador UserController.php que maneja la solicitud de respaldo de la base de datos y ejecuta el método correspondiente del modelo.



```

1  if($controller == 'user'){
2      $controllerPath = __DIR__ . "/app/controllers/UserController.php";
3
4      if (!file_exists($controllerPath)) { die("Error: No se encuentra UserController.php"); }
5
6      include_once $controllerPath;
7      $UserController = new UserController($connection);
8
9      switch($action){
10         case 'insert':
11             $UserController->insertarUsuario();
12             break;
13         case 'consult':
14             $UserController->consultarUsuarios();
15             break;
16         case 'update':
17             $UserController->actualizarUsuario();
18             break;
19         case 'delete':
20             $UserController->eliminarUsuario();
21             break;
22         case 'backup':
23             $UserController->realizarRespaldoBD();
24             break;
25         case 'restore':
26             $UserController->restaurarBD();
27             break;
28         default:
29             $UserController->insertarUsuario();
30     }
31 }
32 }

```

Img. 3 Código del respaldo en Index.php

En la imagen número 3 se muestra el código de la vista Index.php que contiene el botón y la interfaz de usuario para iniciar el proceso de respaldo de la base de datos.

Recuperación de la base de datos

```
1 public function restaurarBD($sqlContent) {
2     $sqlContent = preg_replace('/--.*|(\/*[\s\S]*?\*/)/', '', $sqlContent);
3
4     if ($this->connection->multi_query($sqlContent)) {
5         do {
6             if ($result = $this->connection->store_result()) {
7                 $result->free();
8             }
9             } while ($this->connection->more_results() && $this->connection->next_result());
10        return true;
11    }
12    return false;
13 }
```

Img. 4 Código de Recuperación en UserModel.php

En la imagen número 4 se muestra el código PHP del método de recuperación implementado en UserModel.php, que contiene la funcionalidad para restaurar una base de datos desde un archivo de respaldo.

```
1 public function restaurarBD() {
2     if (isset($_POST['restaurar']) && isset($_FILES['archivo_sql'])) {
3         $tmpName = $_FILES['archivo_sql']['tmp_name'];
4
5         if (is_uploaded_file($tmpName)) {
6             $sqlContent = file_get_contents($tmpName);
7
8             if ($this->model->restaurarBD($sqlContent)) {
9                 echo "<script>alert('Restauración completada con éxito!'); window.location='index.php?controller=auth&action=panelAdministrativo';</script>";
10            } else {
11                echo "<script>alert('Error al restaurar la base de datos. El archivo podría estar dañado.');" window.location='index.php?controller=auth&action=panelAdministrativo';</script>";
12            }
13        } else {
14            echo "<script>alert('Error al subir el archivo.');" window.location='index.php?controller=auth&action=panelAdministrativo';</script>";
15        }
16    } else {
17        // Si entra directo sin POST, redirigir al panel
18        header("Location: index.php?controller=auth&action=panelAdministrativo");
19    }
20 }
```

Img. 5 Código del respaldo en UserController.php

En la imagen número 5 se muestra el código PHP del controlador UserController.php que gestiona la solicitud de restauración de la base de datos y procesa el archivo cargado por el usuario.

```
1  if($controller == 'user'){
2      $controllerPath = __DIR__ . "/app/controllers/UserController.php";
3
4      if (!file_exists($controllerPath)) { die("Error: No se encuentra UserController.php"); }
5
6      include_once $controllerPath;
7      $userController = new UserController($connection);
8
9      switch($action){
10         case 'insert':
11             $userController->insertarUsuario();
12             break;
13         case 'consult':
14             $userController->consultarUsuarios();
15             break;
16         case 'update':
17             $userController->actualizarUsuario();
18             break;
19         case 'delete':
20             $userController->eliminarUsuario();
21             break;
22         case 'backup':
23             $userController->realizarRespaldoBD();
24             break;
25         case 'restore':
26             $userController->restaurarBD();
27             break;
28
29         default:
30             $userController->insertarUsuario();
31     }
32 }
```

Img. 6 Código del respaldo en Index.php

En la imagen número 6 se muestra el código de la vista Index.php que presenta la interfaz para cargar y restaurar un archivo de respaldo de la base de datos.

Reportes en PDF

```
1 public function exportarIndividualPDF() {
2     if (!isset($_GET['estudiante_id'])) { die("Error: Falta ID de estudiante"); }
3
4     $idEstudiante = intval($_GET['estudiante_id']);
5     $est = $this->model->obtenerEstudiante($idEstudiante);
6     $evaluaciones = $this->model->obtenerEvaluacionesEstudiante($idEstudiante);
7
8     if (!$est) { die("Estudiante no encontrado"); }
9
10    $pdf = new FPDF();
11    $pdf->AddPage();
12    $pdf->SetFont('Arial', 'B', 16);
13    $pdf->SetTextColor(74, 30, 111);
14    $pdf->Cell(0, 10, mb_convert_encoding('Reporte Individual de Desempeño', 'ISO-8859-1', 'UTF-
15    8'), 0, 1, 'C');
16    $pdf->SetFont('Arial', '', 10);
17    $pdf->SetTextColor(100);
18    $pdf->Cell(0, 5, 'Generado el: ' . date('d/m/Y H:i'), 0, 1, 'C');
19    $pdf->Ln(10);
20
21    $pdf->SetFont('Arial', 'B', 12);
22    $pdf->SetFillColor(245, 245, 245);
23    $pdf->SetTextColor(0);
24    $pdf->Cell(0, 10, mb_convert_encoding('Información del Estudiante', 'ISO-8859-1', 'UTF-8'),
25    0, 1, 'L', true);
26    $pdf->Ln(2);
27
28    $pdf->SetFont('Arial', 'B', 10);
29    $pdf->Cell(30, 8, 'Nombre:', 0, 0);
30    $pdf->SetFont('Arial', '', 10);
31    $pdf->Cell(80, 8, mb_convert_encoding($est['Nombre'] . ' ' . $est['Apellido'], 'ISO-8859-1',
32    'UTF-8'), 0, 0);
33    $pdf->SetFont('Arial', 'B', 10);
34    $pdf->Cell(25, 8, mb_convert_encoding('Matricula:', 'ISO-8859-1', 'UTF-8'), 0, 0);
35    $pdf->SetFont('Arial', '', 10);
36    $pdf->Cell(30, 8, mb_convert_encoding($est['Matricula'], 'ISO-8859-1', 'UTF-8'), 0, 1);
37    $pdf->SetFont('Arial', 'B', 10);
38    $pdf->Cell(30, 8, 'Grupo:', 0, 0);
39    $pdf->SetFont('Arial', '', 10);
40    $pdf->Cell(80, 8, mb_convert_encoding($est['Grupo'], 'ISO-8859-1', 'UTF-8'), 0, 0);
41    $pdf->Ln(10);
42
43    $pdf->SetFont('Arial', 'B', 11);
44    $pdf->SetFillColor(74, 30, 111);
45    $pdf->SetTextColor(255);
46    $pdf->Cell(30, 10, 'Fecha', 1, 0, 'C', true);
47    $pdf->Cell(100, 10, 'Competencia', 1, 0, 'L', true);
48    $pdf->Cell(35, 10, mb_convert_encoding('Calificación', 'ISO-8859-1', 'UTF-8'), 1, 0, 'C', tru
49    e);
50    $pdf->Cell(25, 10, 'Tipo', 1, 1, 'C', true);
51
52    $pdf->SetFont('Arial', '', 9);
53    $pdf->SetTextColor(0);
54    $pdf->SetFillColor(255);
55
56    foreach ($evaluaciones as $ev) {
57        $pdf->Cell(30, 8, date('d/m/Y', strtotime($ev['Fecha'])), 1, 0, 'C');
58        $pdf->Cell(100, 8, mb_convert_encoding(substr($ev['NombreCompetencia'], 0, 55), 'ISO-8859
59        -1', 'UTF-8'), 1, 0, 'L');
60        $pdf->Cell(35, 8, mb_convert_encoding($ev['Calificacion'], 'ISO-8859-1', 'UTF-8'), 1, 0,
61        'C');
62        $pdf->Cell(25, 8, mb_convert_encoding($ev['TipoEvaluacion'], 'ISO-8859-1', 'UTF-8'), 1,
63        1, 'C');
64    }
65    $pdf->Output('D', 'Reporte_' . $est['Matricula'] . '.pdf');
66 }
```

Img. 7 Código de los reportes en PDF en ReporteController.php

En la imagen número 7 se muestra el código PHP del controlador ReporteController.php que implementa la generación de reportes en formato PDF utilizando una librería de generación de documentos.


```

1  public function obtenerTodosEstudiantes() {
2      $query = "SELECT est.idEstudiantes, est.Matricula, u.Nombre, u.Apellido, est.
Grupo, est.EstadoAcademico
3          FROM estudiantes est INNER JOIN usuarios u ON est.idUsuarios = u.id
Usuarios ORDER BY u.Apellido, u.Nombre";
4      $result = $this->connection->query($query);
5      $data = [];
6      if ($result) while ($row = $result->fetch_assoc()) $data[] = $row;
7      return $data;
8  }
9
10     public function obtenerEstudiante($id) {
11         $query = "SELECT est.idEstudiantes, est.Matricula, u.Nombre, u.Apellido, est.
Grupo, est.EstadoAcademico
12             FROM estudiantes est INNER JOIN usuarios u ON est.idUsuarios = u.id
Usuarios WHERE est.idEstudiantes = ?";
13         $stmt = $this->connection->prepare($query);
14         $stmt->bind_param("i", $id);
15         $stmt->execute();
16         return $stmt->get_result()->fetch_assoc();
17     }
18
19     public function obtenerEvaluacionesEstudiante($id) {
20         $query = "SELECT ev.Fecha, ev.Calificacion, ev.TipoEvaluacion, ev.Observacion
es, c.NombreCompetencia, c.Descripcion as DescripcionCompetencia, CONCAT(u.Nombre, '
', u.Apellido) as NombreDocente
21             FROM evaluaciones ev
22             INNER JOIN competencias c ON ev.idCompetencias = c.idCompetencias
23             INNER JOIN docentes d ON ev.idDocentes = d.idDocentes
24             INNER JOIN usuarios u ON d.idUsuarios = u.idUsuarios
25             WHERE ev.idEstudiantes = ? ORDER BY ev.Fecha DESC";
26         $stmt = $this->connection->prepare($query);
27         $stmt->bind_param("i", $id);
28         $stmt->execute();
29         $result = $stmt->get_result();
30         $data = [];
31         while ($row = $result->fetch_assoc()) $data[] = $row;
32         return $data;
33     }

```

Img. 8 Código de los reportes en PDF en ReporteModel.php

En la imagen número 8 se muestra el código PHP del modelo ReporteModel.php que contiene los métodos para consultar y preparar los datos que se incluirán en los reportes PDF.

```
1  if($controller == 'reporte'){
2      $controllerPath = __DIR__ . "/app/controllers/ReporteController.php";
3
4      if (file_exists($controllerPath)) {
5          include_once $controllerPath;
6          $reporteController = new ReporteController();
7
8          switch($action){
9              // Vistas HTML
10             case 'gruppal':
11                 $reporteController->gruppal();
12                 break;
13             case 'individual':
14                 $reporteController->individual();
15                 break;
16             case 'historico':
17                 $reporteController->historico();
18                 break;
19             case 'exportarIndividualPDF':
20                 $reporteController->exportarIndividualPDF();
21                 break;
22             case 'exportarGruppalPDF':
23                 $reporteController->exportarGruppalPDF();
24                 break;
25             case 'exportarHistoricoPDF':
26                 $reporteController->exportarHistoricoPDF();
27                 break;
28
29             default:
30                 $reporteController->gruppal();
31         }
32     }
```

[Img. 9 Código de los reportes en PDF en index.php](#)

En la imagen número 9 se muestra el código de la vista Index.php que presenta la interfaz de usuario para generar y descargar reportes en formato PDF.

Reportes con gráficas en PDF

```
1 public function exportarGrupalPDF() {
2     $datosGrupos = $this->model->obtenerEstadisticasTodosGrupos();
3
4     $pdf = new FPDF();
5     $pdf->AddPage();
6     $pdf->SetFont('Arial', 'B', 16);
7     $pdf->SetTextColor(247, 148, 29);
8     $pdf->Cell(0, 10, mb_convert_encoding('Reporte Comparativo Grupal', 'ISO-8859-1', 'UTF-8'),
9     0, 1, 'C');
10    $pdf->Ln(5);
11
12    // --- DIBUJAR GRÁFICA ---
13    $this->dibujarGraficaBarras($pdf, $datosGrupos);
14
15    // Mover cursor abajo de la gráfica para la tabla
16    $pdf->SetXY(10, 110);
17
18    // --- TABLA DE DATOS ---
19    $pdf->SetFont('Arial', 'B', 12);
20    $pdf->SetTextColor(0);
21    $pdf->Cell(0, 10, 'Detalle Numérico', 0, 1);
22
23    $pdf->SetFillColor(247, 148, 29);
24    $pdf->SetTextColor(255);
25    $pdf->SetFont('Arial', 'B', 10);
26
27    $w = [30, 35, 35, 30, 30, 30];
28    $pdf->Cell($w[0], 10, 'Grupo', 1, 0, 'C', true);
29    $pdf->Cell($w[1], 10, 'Alumnos', 1, 0, 'C', true);
30    $pdf->Cell($w[2], 10, 'Evaluaciones', 1, 0, 'C', true);
31    $pdf->Cell($w[3], 10, 'Excelentes', 1, 0, 'C', true);
32    $pdf->Cell($w[4], 10, 'Buenas', 1, 0, 'C', true);
33    $pdf->Cell($w[5], 10, 'Bajas', 1, 1, 'C', true);
34
35    $pdf->SetTextColor(0);
36    $pdf->SetFont('Arial', '', 10);
37
38    foreach ($datosGrupos as $grupo) {
39        $pdf->Cell($w[0], 8, $grupo['Grupo'], 'ISO-8859-1', 'UTF-8'), 1, 0,
40        'C');
41        $pdf->Cell($w[1], 8, $grupo['total_estudiantes'], 1, 0, 'C');
42        $pdf->Cell($w[2], 8, $grupo['total_evaluaciones'], 1, 0, 'C');
43        $pdf->Cell($w[3], 8, $grupo['excelentes'], 1, 0, 'C');
44        $pdf->Cell($w[4], 8, $grupo['buenas'], 1, 0, 'C');
45        $pdf->Cell($w[5], 8, $grupo['deficientes'], 1, 1, 'C');
46    }
47
48    $pdf->Output('D', 'Reporte_Grupal.pdf');
```

Img. 10 Código de los reportes con grafica en PDF en ReporteController.php

En la imagen número 10 se muestra el código PHP del controlador ReporteController.php que genera reportes PDF con gráficas estadísticas integradas.

```

1  public function obtenerEstadisticasTodosGrupos() {
2      $grupos = [];
3      $queryGroups = "SELECT DISTINCT Grupo FROM estudiantes ORDER BY Grupo";
4      $resultGroups = $this->connection->query($queryGroups);
5
6      if ($resultGroups) {
7          while ($row = $resultGroups->fetch_assoc()) {
8              $grupo = $row['Grupo'];
9
10             $queryStats = "
11                 SELECT
12                     e.Grupo,
13                     COUNT(DISTINCT est.idEstudiantes) as total_estudiantes,
14                     COUNT(ev.idEvaluaciones) as total_evaluaciones,
15                     SUM(CASE WHEN ev.Calificacion = 'Excelente' THEN 1 ELSE 0 EN
16 D) as excelentes,
17                     SUM(CASE WHEN ev.Calificacion = 'Buena' OR ev.Calificacion =
18 'Bueno' THEN 1 ELSE 0 END) as buenas,
19                     SUM(CASE WHEN ev.Calificacion = 'Deficiente' THEN 1 ELSE 0 EN
20 D) as deficientes
21                 FROM estudiantes e
22                 LEFT JOIN evaluaciones ev ON e.idEstudiantes = ev.idEstudiantes
23                 LEFT JOIN estudiantes est ON e.Grupo = est.Grupo
24                 WHERE e.Grupo = ?
25                 GROUP BY e.Grupo
26             ";
27
28             $stmt = $this->connection->prepare($queryStats);
29             $stmt->bind_param("s", $grupo);
30             $stmt->execute();
31             $resStats = $stmt->get_result();
32
33             if ($data = $resStats->fetch_assoc()) {
34                 $grupos[] = $data;
35             }
36         }
37     }
38     return $grupos;
39 }

```

Img. 11 Código de los reportes con gráfica en PDF en ReporteModel.php

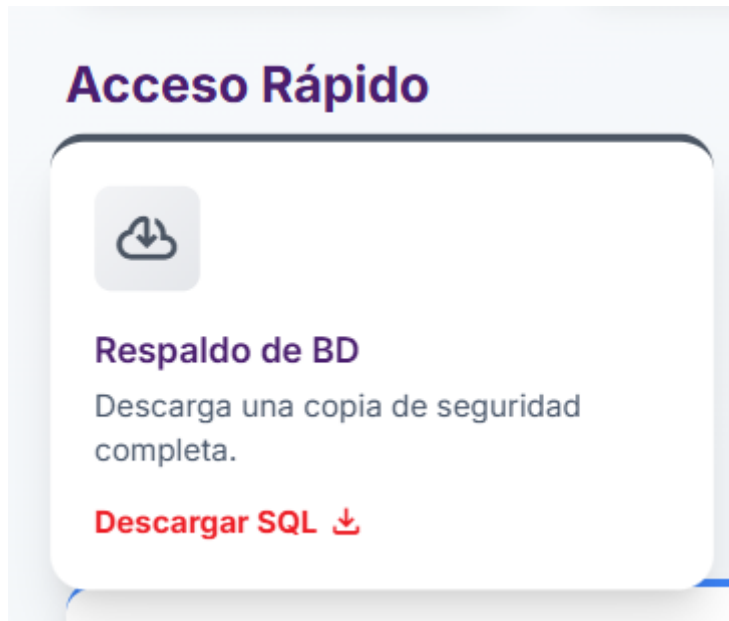
En la imagen número 11 se muestra el código PHP del modelo ReporteModel.php que prepara los datos necesarios para la generación de gráficas en los reportes PDF.

```
1  if($controller == 'reporte'){
2      $controllerPath = __DIR__ . "/app/controllers/ReporteController.php";
3
4      if (file_exists($controllerPath)) {
5          include_once $controllerPath;
6          $reporteController = new ReporteController();
7
8          switch($action){
9              // Vistas HTML
10             case 'gruppal':
11                 $reporteController->gruppal();
12                 break;
13             case 'individual':
14                 $reporteController->individual();
15                 break;
16             case 'historico':
17                 $reporteController->historico();
18                 break;
19             case 'exportarIndividualPDF':
20                 $reporteController->exportarIndividualPDF();
21                 break;
22             case 'exportarGruppalPDF':
23                 $reporteController->exportarGruppalPDF();
24                 break;
25             case 'exportarHistoricoPDF':
26                 $reporteController->exportarHistoricoPDF();
27                 break;
28
29             default:
30                 $reporteController->gruppal();
31         }
32     }
```

[Img. 12 Código de los reportes con gráfica en PDF en Index.php](#)

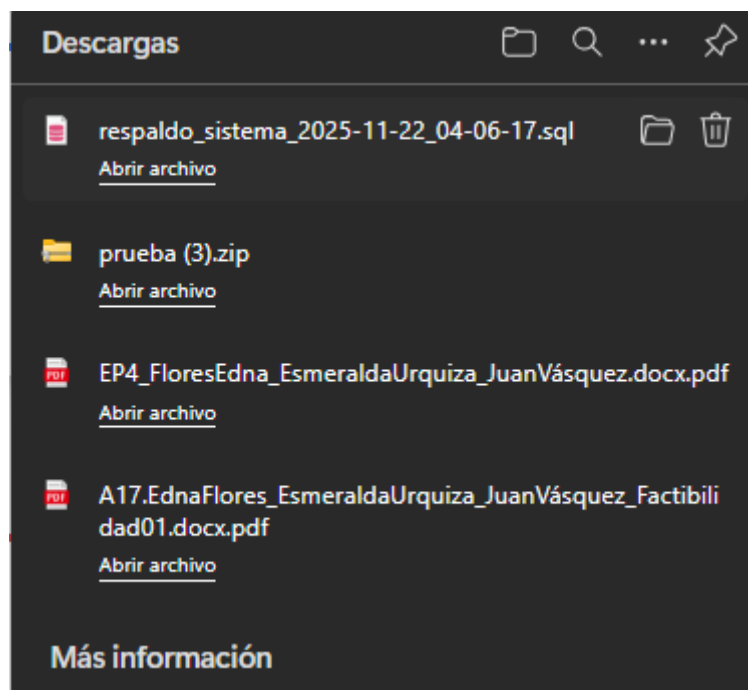
En la imagen número 12 se muestra el código de la vista Index.php con la interfaz para generar reportes PDF que incluyen representaciones gráficas de los datos.

Pruebas



Img. 13 botón de respaldo

En la imagen número 13 se muestra la captura de pantalla de la interfaz web mostrando el botón para iniciar el proceso de respaldo de la base de datos.



Img. 14 descarga de respaldo

En la imagen número 14 se muestra la captura de pantalla del navegador mostrando la descarga del archivo de respaldo de la base de datos en formato SQL.



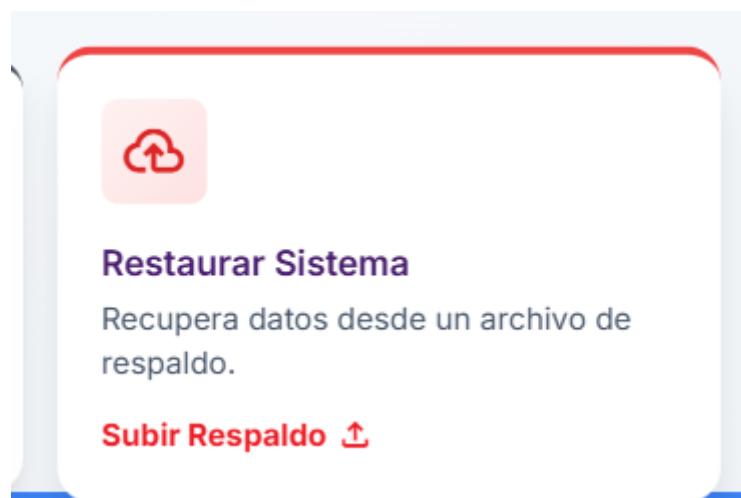
Img. 15 base de datos

En la imagen número 15 se muestra la captura de pantalla de la base de datos completa con todas sus tablas y registros visualizados en phpMyAdmin.



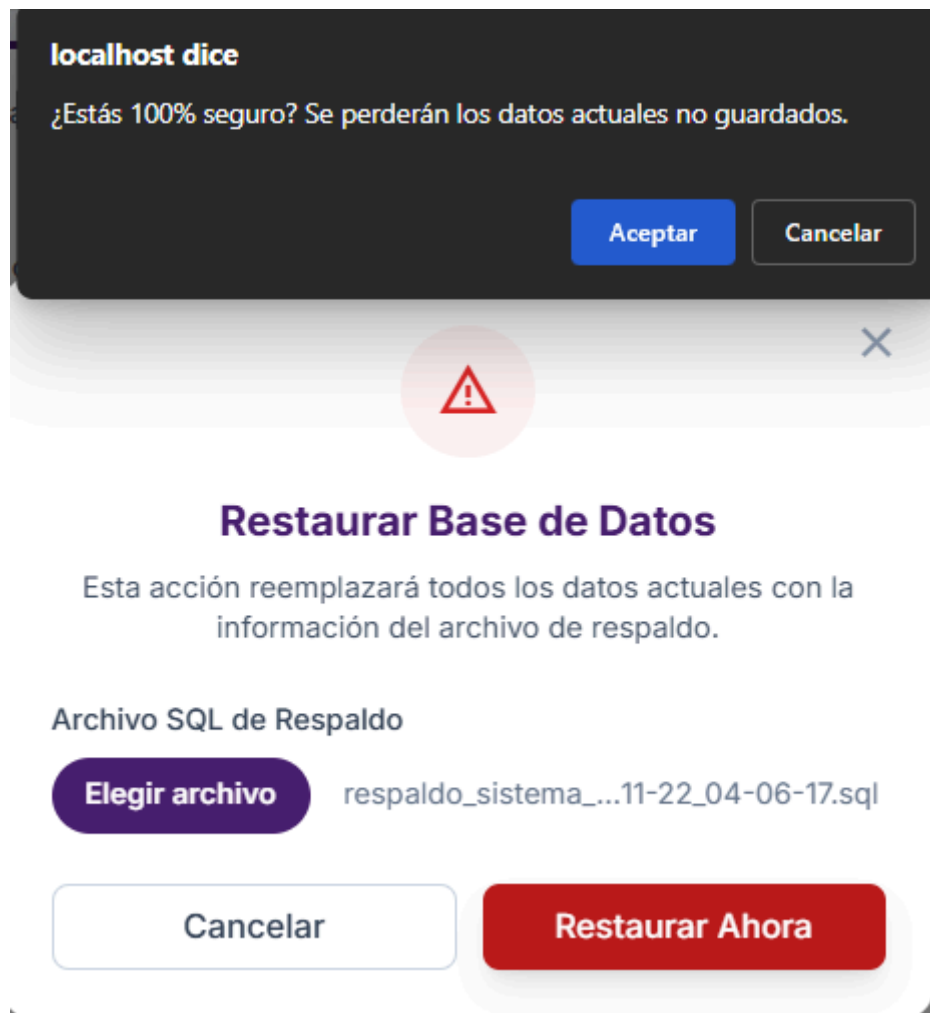
Img. 16 base de datos incompleta

En la imagen número 16 se muestra la captura de pantalla de la base de datos en estado incompleto, con tablas o registros faltantes para demostrar la necesidad de restauración.



Img. 17 botón de la restauración

En la imagen número 17 se muestra la captura de pantalla de la interfaz web con el botón de restauración y el selector de archivos para cargar el respaldo.



Img. 18 archivo de respaldo

En la imagen número 18 se muestra la captura de pantalla del explorador de archivos del sistema mostrando el archivo de respaldo SQL seleccionado para la restauración. Además un mensaje emergente para la verificación de los cambios a hacer en el sistema.



Img. 19 base de datos restaurada

En la imagen número 19 se muestra la captura de pantalla de la base de datos después del proceso de restauración, mostrando todas las tablas y registros recuperados correctamente.

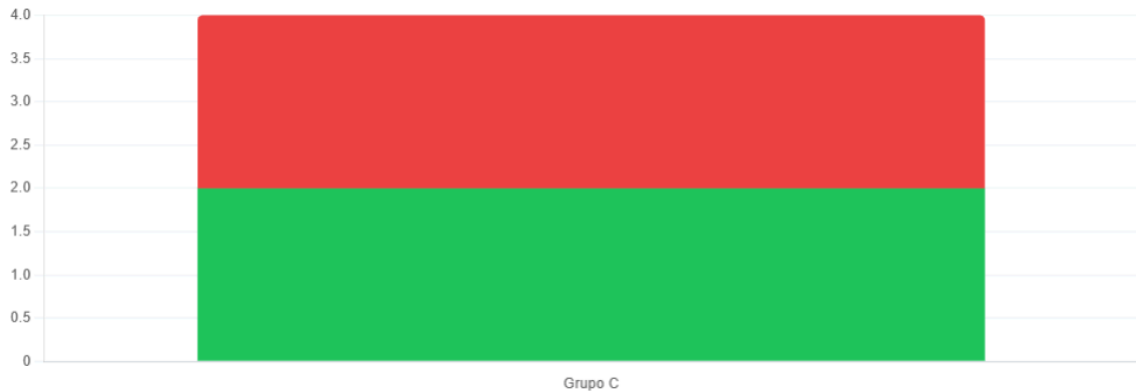
Reporte Grupal

Comparativa de rendimiento académico por grupos

 Exportar PDF

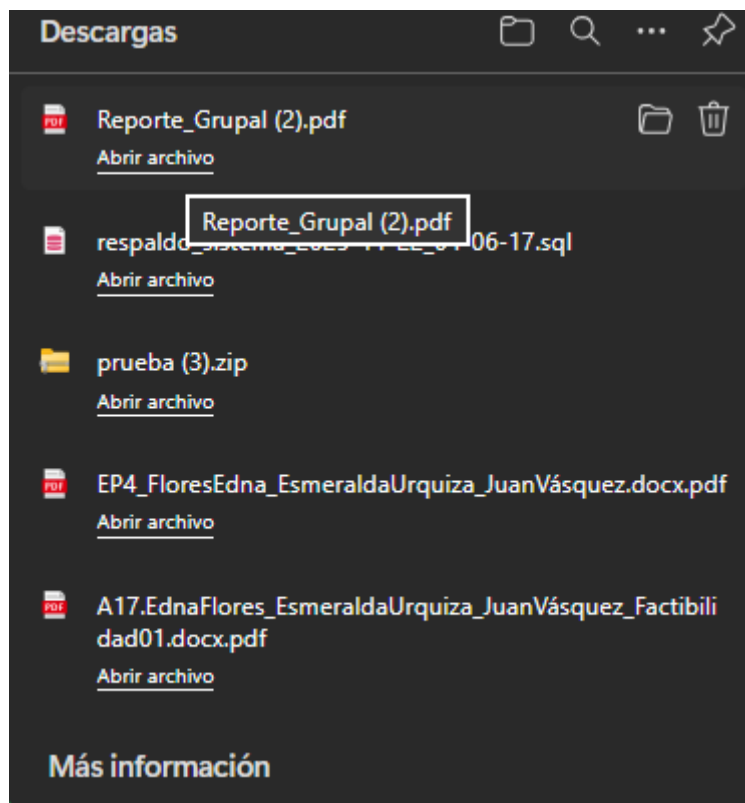
Desempeño por Grupo

● Excelente ● Buena ● Deficiente



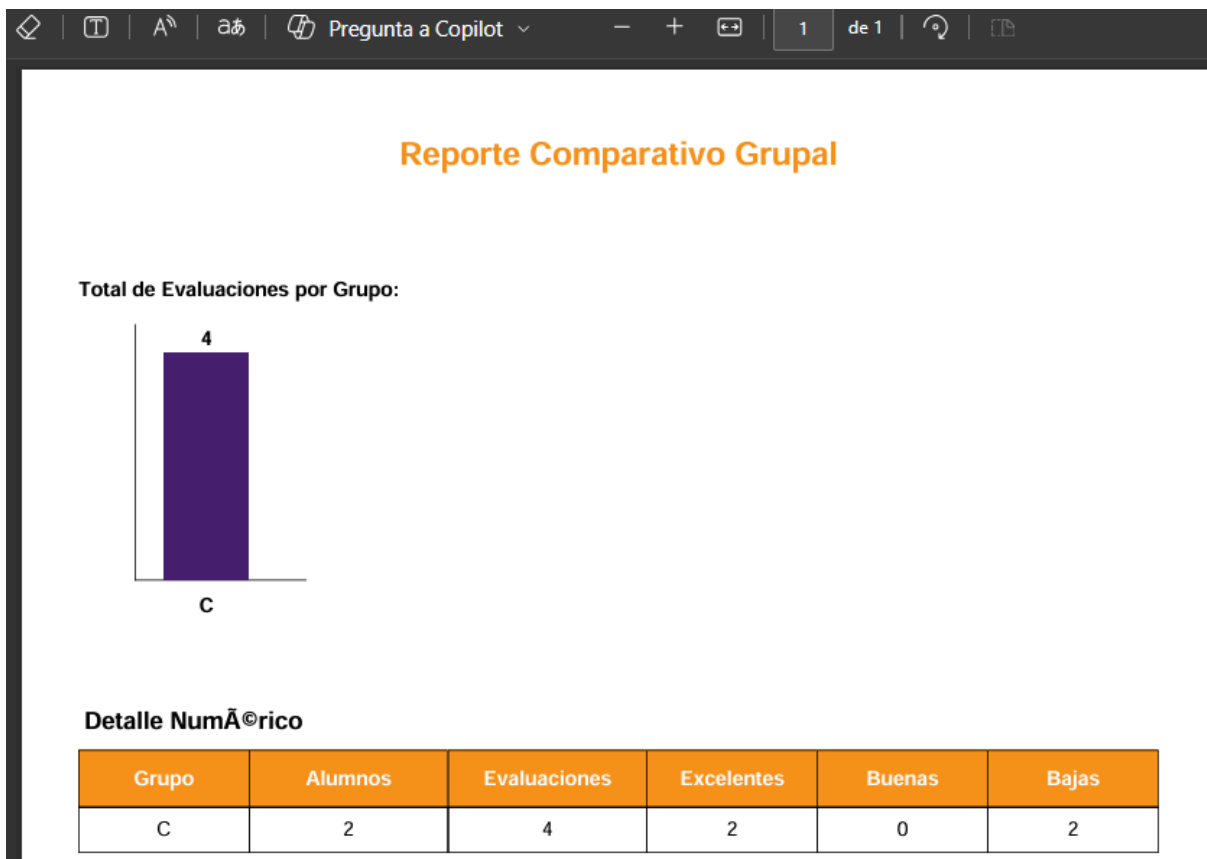
Img. 20 reporte grupal en página web

En la imagen número 20 se muestra la captura de pantalla de un reporte grupal visualizado directamente en la página web del sistema antes de su conversión a PDF.



Img. 21 descarga del reporte con gráfica en pdf

En la imagen número 21 se muestra la captura de pantalla del navegador mostrando la descarga del archivo PDF del reporte que incluye gráficas estadísticas.



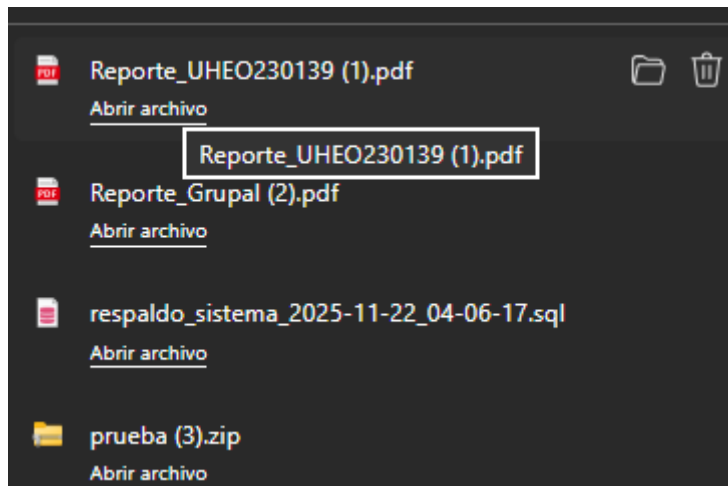
[Img. 22 reporte con gráfica en pdf](#)

En la imagen número 22 se muestra la captura de pantalla del reporte PDF abierto, mostrando las gráficas estadísticas y la visualización de datos de forma gráfica.



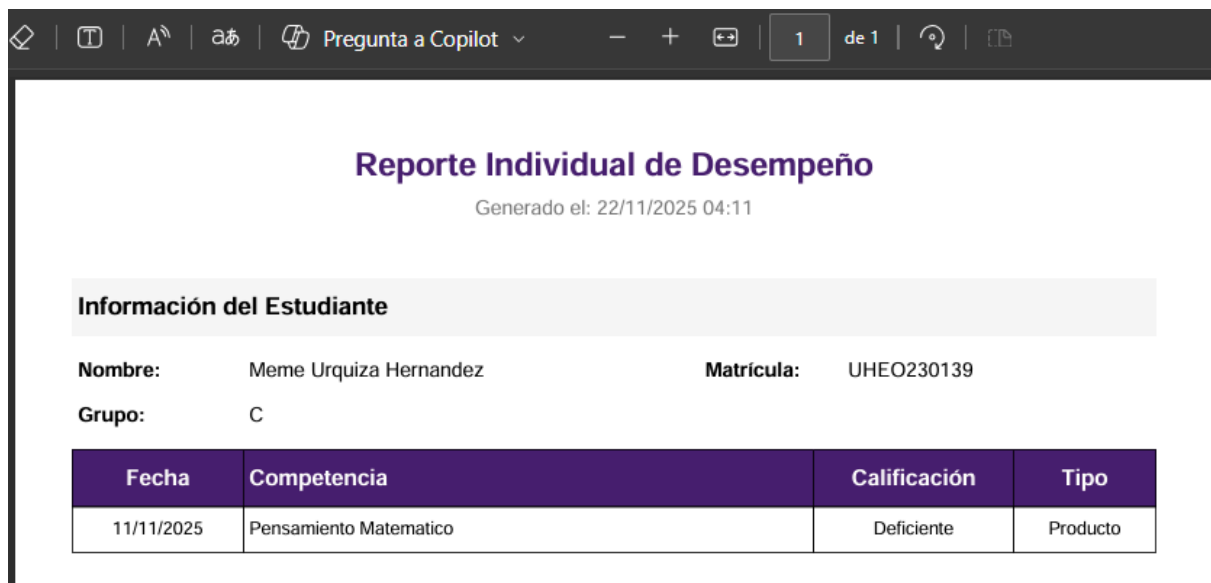
[Img. 23 reporte en página web](#)

En la imagen número 23 se muestra la captura de pantalla de un reporte visualizado en la página web del sistema con tablas de datos y formato estructurado.



Img. 24 descarga del reporte en pdf

En la imagen número 24 se muestra la captura de pantalla del navegador mostrando la descarga del archivo PDF del reporte estándar sin gráficas.



Img. 25 reporte en pdf

En la imagen número 25 se muestra la captura de pantalla del reporte PDF abierto, mostrando el formato final del documento con los datos tabulados.

Conclusiones

La implementación de la aplicación web para la gestión de competencias ha demostrado ser exitosa en los tres componentes fundamentales desarrollados. El módulo de respaldo permite generar copias de seguridad de la base de datos de manera sencilla y confiable, generando archivos SQL que contienen toda la estructura y datos del sistema.

El sistema de recuperación ha probado su efectividad al restaurar completamente bases de datos desde archivos de respaldo, como se evidenció en las pruebas donde se recuperó exitosamente información previamente eliminada. Esta funcionalidad garantiza la continuidad operativa del sistema ante posibles pérdidas de datos.

Por su parte, el generador de reportes en PDF ofrece dos modalidades de gran utilidad: reportes estándar con datos tabulados y reportes con gráficas estadísticas integradas. Ambas opciones facilitan el análisis y la presentación de información de manera profesional y estructurada.

La arquitectura MVC utilizada ha permitido mantener un código organizado, modular y fácil de mantener, separando claramente la lógica de negocio (Modelo), el control de flujo (Controlador) y la presentación (Vista). Las pruebas realizadas confirman que todas las funcionalidades operan correctamente y cumplen con los requerimientos establecidos.

Este sistema proporciona herramientas esenciales para la administración de datos, asegurando tanto la protección de la información como su correcta visualización y análisis, elementos clave para cualquier aplicación web de nivel empresarial.