Proyecto DWS 2^a EVAL

Marcos Martínez Iván García Iris McKirdy

ASPECTOS TÉCNICOS

ARQUITECTURA: Hexagonal

BASE DE DATOS: MariaDB (en Docker)

LENGUAJE DE PROGRAMACIÓN: JAVA

FRAMEWORK: SpringBoot

LIBRERÍAS

ApachePoi: generación de Excel.

JDBC: acceso a la base de datos.

Lombok: facilitar el desarrollo del código.

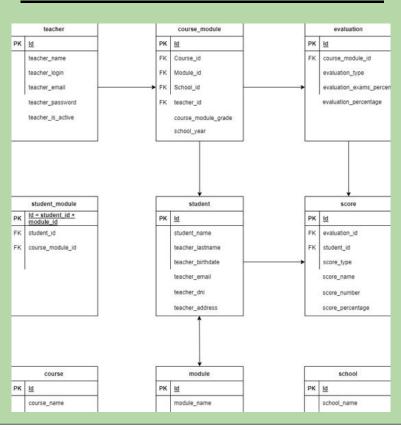
MapStruct: mapeo de bbdd a entidades.

MySqlConnector: conexión a la bbdd.

Thymeleaf: conexión con el front-end.

SODS: parseo de Excel.

MAPA ENTIDAD-RELACIÓN



ARQUITECTURA

Capas:

gno-g2-api-rest

gno-g2-application

gno-g2-boot

gno-g2-domain

gno-g2-infrastructure

INICIAR LA BBDD CON DOCKER

```
Dokcerfile:
  FROM mariadb:latest
  ADD ddl/ /docker-entrypoint-initdb.d
  ENV MARIADB USER=usuario1
  ENV MARIADB_PASSWORD=abc123.
  ENV MARIADB ROOT PASSWORD=abc123.
  EXPOSE 3306
```

INICIAR LA BBDD CON DOCKER

- 1. Abrir docker
- 2. Abrir la consola de comandos y ejecutar el comando (para construir la imagen):
 - docker build rutaDirecotioDelDockerfile
- 3. Levantar el contenedor:
 - docker run -p 666:3306 -v mariadb_data:/var/lib/mariadb idDeLaImagen

LOG-IN

URL: localhost:8081/login

Username: guc Password:		Login	
Password:	quc	Username:	
Submit		Password:	
Cubinit.		Submit	

LOG-IN

Profesores registrados:

usuario: flr

constraseña: abc123.

usuario: guc constraseña: abc123.

usuario: pma constraseña: abc123.

API Enviar Excel POST Endpoint: /excel/read Body: archivo Excel

```
@PostMapping(value = "/read")
public ResponseEntity<String> readExcel(@RequestParam("file") MultipartFile file) throws IOException {
    try {
        InputStream inputStream = file.getInputStream();
        SpreadSheet spread = new SpreadSheet(inputStream);
       List<Sheet> sheets = spread.getSheets();
        for (Sheet sheet : sheets) {
           // System.out.println("In sheet " + sheet.getName());
            Range range = sheet.getDataRange();
           //System.out.println(range.getCell(1,1).getValue().toString());
           service.postExcel(file);
    } catch (Exception e){
        return new ResponseEntity<>(e.getMessage(), HttpStatus.BAD_REQUEST);
    HttpHeaders headers = new HttpHeaders();
   headers.setLocation(URI.create("/index"));
   return new ResponseEntity<>(headers, HttpStatus.FOUND);
```

API Enviar Excel POST Endpoint: /excel/read Body: archivo Excel

REQUERIMIENTOS DEL EXCEL

1. Celda en la primera fila con el porcentaje de las tareas, exámenes y evaluación:

PTE_EVALx_tareasx_examenx_totalx

2. Celda en la primera fila con el año, centro, profesor, curso y módulo: ACPCM_yyyy_centro_emailprofesor_curso_módulo

<u>API</u>

Recibir Excel (desde el front) GET Endpoint: /excel/Parámetros: ciclo (nombre), módulo(nombre), año escolar.

```
@GetMapping()
public ResponseEntity<XSSFWorkbook> getExcel(HttpServletResponse response, @RequestParam String course, @RequestParam
    response.setContentType("application/octet-stream"):
    DateFormat dateFormatter = new SimpleDateFormat("yyyy-MM-dd_HH:mm:ss");
    String currentDateTime = dateFormatter.format(new Date());
    String headerKey = "Content-Disposition":
    String headerValue = "attachment: filename=notas_" + currentDateTime + ".xlsx";
    response.setHeader(headerKey, headerValue);
   XSSFWorkbook excel = null;
    try {
        excel = getExcelService.getExcel(course, module, schoolYear, response);
    }catch (Exception e){
        System.out.println(e.getMessage());
        System.out.println("Error al generar el Excel");
        return new ResponseEntity<>(HttpStatus.BAD_REQUEST);
    return new ResponseEntity<XSSFWorkbook>(excel, HttpStatus.OK);
```

API

Recibir Excel (desde postman) GET Endpoint: /excel/postman Parámetros: ciclo (id), módulo(id), año escolar.

```
@GetMapping(value = "/postman", produces = MediaType.APPLICATION_OCTET_STREAM_VALUE)
public XSSFWorkbook getExcelPostman(HttpServletResponse response, @RequestParam Integer courseId, @RequestParam Intege
    response.setContentType("application/octet-stream");
   DateFormat dateFormatter = new SimpleDateFormat("yyyy-MM-dd_HH:mm:ss");
    String currentDateTime = dateFormatter.format(new Date());
    String headerKey = "Content-Disposition";
    String headerValue = "attachment; filename=notas_" + currentDateTime + ".xlsx";
    response.setHeader(headerKey, headerValue);
    XSSFWorkbook excel = null:
   try {
        excel = getExcelService.getExcelPostman(courseId, moduleId, schoolYear, response);
    }catch (Exception e){
        System.out.println(e.getMessage());
        System.out.println("Error al generar el Excel");
    return excel:
```

GRACIAS POR VUESTRA ATENCIÓN