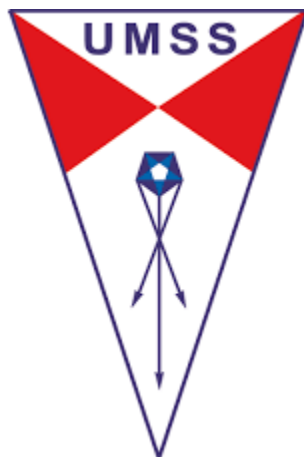


UNIVERSIDAD MAYOR DE SAN SIMÓN  
FACULTAD DE TECNOLOGÍA



INFORME 5  
SEGUNDO HITO - OnDesK  
GRUPO 2

Integrantes:

David Eduardo Chávez Totorá

Maria Clara Santos Marca

Cesar Eduardo Nick Ochoa Beltran

Sergio Andrés Maldonado Sejas

Materia: TALLER DE BASE DE DATOS

12 de noviembre, 2025

Cochabamba - Bolivia

## Estructura del proyecto

La estructura de directorios ideal empleada para llevar a cabo OnDesk se basa en la siguiente estructura:

```
-OnDesk
├── bd
│   ├── tbd_default.sql
│   └── tbd.sql
├── css
│   ├── admin.css
│   ├── administrador.css
│   ├── estudiante.css
│   ├── generic.css
│   ├── global.css
│   ├── inicio.css
│   ├── login.css
│   └── maestro.css
├── img
│   ├── admin.png
│   ├── flechita.png
│   ├── fondo.png
│   ├── logo.png
│   ├── logof.png
│   └── logoW.png
├── js
│   ├── admin.js
│   ├── administrador.js
│   ├── estudiante.js
│   ├── inicio.js
│   ├── login.js
│   ├── loginMaestro.js
│   └── maestro.js
├── libs
│   └── flatpickr
│       ├── es.js
│       ├── flatpickr.min.css
│       └── flatpickr.min.js
├── php
│   └── tests
│       ├── crear_areas.php
│       ├── crear_categorias.php
│       ├── crear_cursos.php
│       ├── crear_grados.php
│       ├── crear_permiso.php
│       └── test.php
├── adminIngresar.php
├── areaGetAll.php
├── areaNew.php
├── becaNew.php
├── becaGetAll.php
├── conexion.php
├── cursoDelete.php
└── cursoEstudianteGet.php
```

```

|   | cursoGetAll.php
|   | cursoNew.php
|   | estudianteGet.php
|   | estudianteGetAll.php
|   | estudianteIngresar.php
|   | estudianteRanking.php
|   | estudianteRegistrar.php
|   | evaluacionGetByModulo.php
|   | evaluacionNew.php
|   | gradoGetAll.php
|   | maestroIngresar.php
|   | maestroNew.php
|   | maestroGet.php
|   | moduloGetByPc.php
|   | moduloNew.php
|   | maestroIngresar.php
|   | pcAsignarMaestro.php
|   | pcGetByMaestro.php
|   | rankingGetAll.php
|   | rolEditar.php
|   | rolEliminar.php
|   | rolGetAll.php
|   | rolGetFil.php
|   | tareaGetByModulo.php
|   | tareaNew.php
|   | temasGetByModulo.php
|   | temasNew.php
|   └─ admin.html
|   └─ administrador.html
|   └─ estudiante.html
|   └─ inicio.html
|   └─ login.html
|   └─ loginMaestro.html
|   └─ maestro.html

```

La estructura de directorios presentada para OnDesk refleja una arquitectura organizada y escalable que sigue mejores prácticas de desarrollo web. Esta organización sistemática ofrece múltiples ventajas:

### **1. Separación de Responsabilidades (SoC)**

La división en carpetas específicas (css/, js/, php/, bd/, img/) permite que diferentes equipos trabajen simultáneamente sin conflictos y facilita el mantenimiento. Cada capa de la aplicación tiene su espacio definido.

### **2. Escalabilidad y Mantenibilidad**

- Estructura modular: Cada funcionalidad tiene sus archivos específicos
- Fácil localización: Los desarrolladores pueden encontrar rápidamente cualquier recurso
- Actualizaciones seguras: Modificar un módulo no afecta a los demás

### 3. Gestión de Dependencias

- La carpeta libs/ centraliza dependencias externas como flatpickr, facilitando:
- Control de versiones
- Actualizaciones
- Eliminación de duplicados

### 4. Seguridad Estructural

- Separación clara entre frontend y backend
- Los archivos PHP sensibles están aislados
- Los recursos estáticos están organizados por tipo

### 5. Flujo de Desarrollo Eficiente

La organización de la carpeta php/ con archivos como conexion.php, getMaestros.php e ingresarAdm.php refleja un flujo **Modelo Vista Controlador** implícito que mejora la productividad.

### 6. Optimización de Rendimiento

- Los archivos CSS y JS están organizados por páginas
- Facilita la implementación de estrategias de caché
- Permite carga selectiva de recursos

## Cronograma

El cronograma de desarrollo de **OnDesk** para el mes de Octubre de este año obedece a una división de tareas semanales entre los 4 miembros del equipo de desarrollo. Teniendo en cuenta los módulos planteados en el **Hito 1** y los entregables planteados en el **Hito 2** se ha modelado un marco de entregas adecuado que obedezca a las entregas funcionales relacionadas con la base de datos y los módulos básicos para el funcionamiento de la misma.

Entregable	Tareas	Encargado de Tarea
<b>Semana 1</b>	1- Desarrollo de Mocks e Interfaces preliminares 2- Investigación de uso de MySQL 3- Análisis de la Estructura de Datos 4- Investigación y uso de Triggers para OnDesk	1- Cesar Ochoa 2- Sergio Maldonado 3- David Chavez 4- Maria Santos
<b>Semana 2</b>	1- Interfaces + Conexiones 1.1- <b>Desarrollo de interfaces acordadas</b> <ul style="list-style-type: none"><li>• Vista de Inicio</li><li>• Vista de Login Estudiante</li><li>• Vista de Login Administrador</li><li>• Vistas administrador para gestión de</li></ul>	1- Sergio Maldonado 2- Cesar Ochoa 3- David Chávez - Maria Santos

	<p>Trabajadores (Creación, Consulta de Datos)</p> <ul style="list-style-type: none"> <li>• Vistas administrador para gestión de Roles</li> </ul> <p><b>1.2- Integración con Scripts de PHP/Backend</b></p> <p><b>2- Desarrollo de operaciones CRUD (scripts) básicas para roles y Administradores</b></p> <ul style="list-style-type: none"> <li>• Script registrar Trabajadores (Maestros)</li> <li>• Script obtener Maestros</li> <li>• Scripts para roles</li> <li>• Script de conexión a la Base de Datos</li> <li>• Scripts estudiantes básico</li> </ul> <p><b>3- Desarrollo de Scripts varios</b></p> <ul style="list-style-type: none"> <li>• Script conexión con BD</li> <li>• Verificación de LOGIN de administrador</li> </ul>	
<b>Semana 3</b>	<p>1- Desarrollo de interfaces:</p> <ul style="list-style-type: none"> <li>• <b>Vistas desde Administrador</b> <ul style="list-style-type: none"> <li>○ Para registrar/gestionar cursos (Virtuales/Presenciales)</li> <li>○ Para registrar otras áreas.</li> <li>○ Para ver estudiantes registrados y sus detalles.</li> </ul> </li> <li>• <b>Vistas desde Maestro</b> <ul style="list-style-type: none"> <li>○ Del curso visto como Maestro</li> </ul> </li> <li>• <b>Vistas desde Estudiante</b> <ul style="list-style-type: none"> <li>○ Menú</li> <li>○ Inscripción a Cursos</li> <li>○ De los cursos inscritos</li> <li>○ Del progreso de sus cursos</li> <li>○ Evaluaciones pautadas (Tareas, Exámenes)</li> </ul> </li> </ul> <p>2- Integración con Scripts de PHP/Backend</p> <ul style="list-style-type: none"> <li>• Scripts para crear estudiantes</li> <li>• Scripts para consultar estudiantes</li> <li>• Scripts para crear cursos</li> <li>• Scripts para borrar cursos</li> <li>• Scripts para consultar cursos</li> <li>• Scripts para crear maestros</li> <li>• Scripts para consultar maestros</li> </ul> <p>3- Desarrollo de Scripts: Datos predeterminados</p> <ul style="list-style-type: none"> <li>• Script preliminar de la base de datos ya cargada (Rankings, Categorías, Grados, Roles)</li> </ul>	<p>1-Cesar Ochoa - Sergio Maldonado</p> <p>2-David Chavez</p> <p>3-Maria Santos</p>
<b>Semana 4</b>	<p>1- Desarrollo de interfaces</p> <ul style="list-style-type: none"> <li>• <b>Vistas desde Administrador</b> <ul style="list-style-type: none"> <li>○ Revisar y aprobar solicitudes de maestros para impartir cursos (Pendiente / Aceptado / Rechazado).</li> <li>○ Gestionar asignación de aulas para los periodos de cursos presenciales.</li> <li>○ Revisar reportes de cupos ocupados y solicitudes totales de cada curso.</li> <li>○ Dar becas a estudiantes según área y porcentaje definido, registrando fechas de inicio y fin.</li> <li>○ Definir descuentos disponibles para cursos, incluyendo porcentaje y fechas de vigencia.</li> <li>○ Visualizar estadísticas de cursos: recaudación,</li> </ul> </li> </ul>	<p>1- Cesar Ochoa - Sergio Maldonado</p> <p>2- David Chavez</p> <p>3- Maria Santos</p>

	<p>cupos ocupados y disponibles.</p> <ul style="list-style-type: none"> <li>○ Actualizar información de roles y permisos si es necesario.</li> <li>○ Gestión de Becas y Descuentos (registro, porcentaje, estado y área )</li> <li>○ Vistas de Control de aulas y asistencias .</li> <li>○ Revisión y edición de cursos, maestros y estudiantes .</li> <li>○ Para asignar Maestros a cursos</li> <li>○ Para consulta de Rankings</li> </ul> <p>● <b>Vistas desde Maestro</b></p> <ul style="list-style-type: none"> <li>○ Ingresar fecha de inicio y fin del periodo de su curso.</li> <li>○ Definir cantidad de cupos disponibles y calcular cupos ocupados y solicitudes.</li> <li>○ Ingresar costo del curso y calcular recaudación según cupos ocupados.</li> <li>○ Definir horario del curso y verificar capacidad del aula (si es presencial).</li> <li>○ Crear módulos y tareas dentro del periodo del curso: títulos, descripciones, fechas y horas.</li> <li>○ Evaluar tareas y exámenes del estudiante, asignando notas y puntos de premio.</li> <li>○ Ingresar material y temas para el curso, incluyendo archivos adjuntos opcionales y comentarios públicos.</li> <li>○ Del perfil de Maestro (actualización de cursos impartidos )</li> <li>○ Vistas de evaluaciones y retroalimentaciones.</li> <li>○ Opciones para definir estructuras de curso (Módulos, Material)</li> <li>○ Opciones para agregar evaluaciones (Tareas, Exámenes)</li> </ul> <p>● <b>Vistas desde Estudiante</b></p> <ul style="list-style-type: none"> <li>○ Inscribirse a un periodo de curso según su grado y disponibilidad.</li> <li>○ Realizar entregas de tareas y evaluaciones, registrando fecha y hora de entrega.</li> <li>○ Visualizar horarios de cursos, aulas o modalidad virtual.</li> <li>○ Registrar su asistencia en el curso.</li> <li>○ Revisar estadísticas del periodo: nota final, asistencias, puntos de recompensa y ranking.</li> <li>○ Del perfil de Estudiante ( grado , recompensas , insignias , puntos )</li> <li>○ Vista de Tienda de cosméticos y temas del sistema de gamificación.</li> </ul> <p>2- Integración general y funcionalidades</p> <ul style="list-style-type: none"> <li>● Integrar la funcionalidad de Becas y Descuentos activos.</li> <li>● Integrar el sistema de gamificación (rankings, insignias, recompensas, puntos)</li> </ul> <p>3- Pruebas y Seguridad</p> <ul style="list-style-type: none"> <li>● Verificación de registro y roles (Administrador, Maestro y Estudiante)</li> <li>● Pruebas de conexión con la base de datos y triggers.</li> <li>● Control de seguridad )</li> <li>● Pruebas de estrés de la base de datos</li> </ul>	
--	---	--

### SCRIPT (BD)

```
-- Estructura de tabla para la tabla `archivos_adjuntos`
CREATE TABLE `archivos_adjuntos` (
  `id_archivo` int(11) NOT NULL,
  `titulo` varchar(61) DEFAULT NULL,
  `tipo` varchar(50) DEFAULT NULL,
  `ruta_archivo` varchar(255) DEFAULT NULL,
  `fecha_subida` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `archivos_publicacion`
CREATE TABLE `archivos_publicacion` (
  `id_archivo` int(11) NOT NULL,
  `id_publicacion` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `area`
CREATE TABLE `area` (
  `id_area` int(11) NOT NULL,
  `nombre_area` varchar(31) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `aula`
CREATE TABLE `aula` (
  `id_aula` int(11) NOT NULL,
  `capacidad` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `bitacora_inscripcion_estudiante` TRIGGER
CREATE TABLE `bitacora_inscripcion_estudiante` (
  `id_bitInscripcionEstudiante` int(11) NOT NULL,
  `accion` varchar(50) DEFAULT NULL,
  `fecha` datetime DEFAULT NULL,
  `descripcion` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `categoria`
CREATE TABLE `categoria` (
  `id_categoria` int(11) NOT NULL,
  `nombre_categoria` varchar(31) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `comentario`
CREATE TABLE `comentario` (
  `id_comentario` int(11) NOT NULL,
  `id_publicacion` int(11) DEFAULT NULL,
  `mensaje` varchar(1001) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `cosmetico`
CREATE TABLE `cosmetico` (
  `id_cosmetico` int(11) NOT NULL,
  `id_tipo_cosmetico` int(11) DEFAULT NULL,
  `costo_canje` int(11) DEFAULT NULL
```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `curso`
CREATE TABLE `curso` (
  `id_curso` int(11) NOT NULL,
  `id_categoria` int(11) DEFAULT NULL,
  `id_area` int(11) DEFAULT NULL,
  `id_grado` int(11) DEFAULT NULL,
  `duracion` int(11) DEFAULT NULL,
  `titulo` varchar(31) DEFAULT NULL,
  `modalidad` varchar(2) DEFAULT NULL,
  `inicio_gestion` date DEFAULT NULL,
  `fin_gestion` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `curso_estudiante`
CREATE TABLE `curso_estudiante` (
  `id_curso_estudiante` int(11) NOT NULL,
  `id_estudiante` int(11) DEFAULT NULL,
  `id_periodo_curso` int(11) DEFAULT NULL,
  `estado` varchar(21) DEFAULT NULL,
  `nota` int(11) DEFAULT NULL,
  `asistencia` int(11) DEFAULT NULL,
  `deskPoints` int(11) DEFAULT NULL,
  `rankingPoints` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `curso_maestro`
CREATE TABLE `curso_maestro` (
  `id_curso_maestro` int(11) NOT NULL,
  `id_maestro` int(11) DEFAULT NULL,
  `id_periodo_curso` int(11) DEFAULT NULL,
  `pago_maestro` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `datos_maestro`
CREATE TABLE `datos_maestro` (
  `id_dato` int(11) NOT NULL,
  `id_user` int(11) DEFAULT NULL,
  `titulo` varchar(51) DEFAULT NULL,
  `sueldo` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `descuento`
CREATE TABLE `descuento` (
  `id_descuento` varchar(11) NOT NULL,
  `id_periodo_curso` int(11) DEFAULT NULL,
  `costo_canje` int(11) DEFAULT NULL,
  `fecha_fin` date DEFAULT NULL,
  `porcentaje_descuento` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `dia_horario`
CREATE TABLE `dia_horario` (
  `id_dia_clase` int(11) NOT NULL,

```



```

`dia` varchar(10) DEFAULT NULL,
`hora_inicio` time DEFAULT NULL,
`hora_fin` time DEFAULT NULL,
`id_aula` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `entregas`
CREATE TABLE `entregas` (
  `id_entrega` int(11) NOT NULL,
  `id_user` int(11) DEFAULT NULL,
  `nota` decimal(10,2) DEFAULT NULL,
  `hora_entrega` time DEFAULT NULL,
  `fecha_entrega` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `evaluacion`
CREATE TABLE `evaluacion` (
  `id_evaluacion` int(11) NOT NULL,
  `id_modulo` int(11) DEFAULT NULL,
  `titulo` varchar(51) DEFAULT NULL,
  `descripcion` varchar(1001) DEFAULT NULL,
  `hora_emision` time DEFAULT NULL,
  `fecha_emision` date DEFAULT NULL,
  `hora_inicio` time DEFAULT NULL,
  `fecha_inicio` date DEFAULT NULL,
  `hora_entrega` time DEFAULT NULL,
  `fecha_entrega` date DEFAULT NULL,
  `deskPoints` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `grado`
CREATE TABLE `grado` (
  `id_grado` int(11) NOT NULL,
  `nombre_grado` varchar(31) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `horario`
CREATE TABLE `horario` (
  `id_periodo_clase` int(11) NOT NULL,
  `id_dia_clase` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `inscripcion`
CREATE TABLE `inscripcion` (
  `id_inscripcion` int(11) NOT NULL,
  `id_tipo_pago` int(11) DEFAULT NULL,
  `id_user` int(11) DEFAULT NULL,
  `id_periodo_curso` int(11) DEFAULT NULL,
  `id_descuento` varchar(11) DEFAULT NULL,
  `fecha_inscripcion` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `insignias`
CREATE TABLE `insignias` (
  `id_insignia` int(11) NOT NULL,

```

```

`nombre` varchar(31) DEFAULT NULL,
`descripcion` varchar(101) DEFAULT NULL,
`puntos_premio` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `insignias_estudiantes`
CREATE TABLE `insignias_estudiantes` (
  `id_insignias_estudiantes` int(11) NOT NULL,
  `id_estudiante` int(11) DEFAULT NULL,
  `id_insignia` int(11) DEFAULT NULL,
  `puntos_premio` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `material`
CREATE TABLE `material` (
  `id_material` int(11) NOT NULL,
  `id_tema` int(11) DEFAULT NULL,
  `titulo` varchar(51) DEFAULT NULL,
  `descripcion` varchar(1001) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `modulo`
CREATE TABLE `modulo` (
  `id_modulo` int(11) NOT NULL,
  `id_periodo_curso` int(11) DEFAULT NULL,
  `titulo` varchar(51) DEFAULT NULL,
  `orden` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `periodo_curso`
CREATE TABLE `periodo_curso` (
  `id_periodo_curso` int(11) NOT NULL,
  `id_curso` int(11) DEFAULT NULL,
  `id_maestro` int(11) DEFAULT NULL,
  `fecha_inicio` date DEFAULT NULL,
  `fecha_fin` date DEFAULT NULL,
  `cupos` int(11) DEFAULT NULL,
  `cupos_ocupados` int(11) DEFAULT NULL,
  `solicitudes_totales` int(11) DEFAULT NULL,
  `costo` int(11) DEFAULT NULL,
  `recaudado` int(11) DEFAULT NULL,
  `estado_periodo` varchar(21) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `permiso`
CREATE TABLE `permiso` (
  `id_permiso` int(11) NOT NULL,
  `nombre_permiso` varchar(71) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `puntos`
CREATE TABLE `puntos` (
  `id_puntos` int(11) NOT NULL,
  `id_user` int(11) DEFAULT NULL,
  `puntos_totales` decimal(10,2) DEFAULT NULL,

```

```

    `puntos_gastados` decimal(10,2) DEFAULT NULL,
    `saldo_actual` decimal(10,2) DEFAULT NULL,
    `rankingPoints` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `ranking`
CREATE TABLE `ranking` (
    `ranking` varchar(30) NOT NULL,
    `nombre_ranking` varchar(31) DEFAULT NULL,
    `limite_superior` int(11) DEFAULT NULL,
    `limite_inferior` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `recompensa_canjeada`
CREATE TABLE `recompensa_canjeada` (
    `id_recompensa_canjeada` int(11) NOT NULL,
    `recompensa` varchar(11) NOT NULL,
    `id_estudiante` int(11) DEFAULT NULL,
    `fecha_recompensa` date DEFAULT NULL,
    `hora_recompensa` time DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `rol`
CREATE TABLE `rol` (
    `id_rol` int(11) NOT NULL,
    `nombre_rol` varchar(31) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `rol_permiso`
CREATE TABLE `rol_permiso` (
    `id_rol` int(11) NOT NULL,
    `id_permiso` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `rol_usuario`
CREATE TABLE `rol_usuario` (
    `id_user` int(11) NOT NULL,
    `id_rol` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `tarea`
CREATE TABLE `tarea` (
    `id_tarea` int(11) NOT NULL,
    `id_modulo` int(11) DEFAULT NULL,
    `titulo` varchar(51) DEFAULT NULL,
    `descripcion` varchar(1001) DEFAULT NULL,
    `hora_emision` time DEFAULT NULL,
    `fecha_emision` date DEFAULT NULL,
    `hora_entrega` time DEFAULT NULL,
    `fecha_entrega` date DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `temas`
CREATE TABLE `temas` (
    `id_tema` int(11) NOT NULL,
    `id_modulo` int(11) DEFAULT NULL,

```

```

    `titulo` varchar(51) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `tipo_cosmetico`
CREATE TABLE `tipo_cosmetico` (
  `id_tipo_cosmetico` int(11) NOT NULL,
  `titulo` varchar(31) DEFAULT NULL,
  `descripcion` varchar(1001) DEFAULT NULL,
  `imagen` varchar(99) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `tipo_pago`
CREATE TABLE `tipo_pago` (
  `id_tipo_pago` int(11) NOT NULL,
  `tipo_pago` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
-- Estructura de tabla para la tabla `usuario`
CREATE TABLE `usuario` (
  `id_user` int(11) NOT NULL,
  `nombre` varchar(31) DEFAULT NULL,
  `apellido` varchar(31) DEFAULT NULL,
  `username` varchar(16) DEFAULT NULL,
  `contrasenna` varchar(31) NOT NULL,
  `ci` varchar(11) DEFAULT NULL,
  `telefono` varchar(16) DEFAULT NULL,
  `correo` varchar(101) DEFAULT NULL,
  `edad` varchar(4) DEFAULT NULL,
  `estado` varchar(11) NOT NULL DEFAULT 'Activo',
  `grado` varchar(31) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

## TRIGGERS SCRIPT BD

```
-- Disparadores `beca`
--
DELIMITER $$
CREATE TRIGGER `trg_validar_fechas_beca` BEFORE INSERT ON `beca` FOR EACH ROW BEGIN
    IF NEW.fecha_fin < NEW.fecha_inicio THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'La fecha de fin no puede ser anterior a la fecha de inicio.';
    END IF;
END
$$
DELIMITER ;
-- Disparadores `curso_estudiante`
--
DELIMITER $$
CREATE TRIGGER `inscripcion_estudiante` AFTER INSERT ON `curso_estudiante` FOR EACH ROW BEGIN
    DECLARE costo DECIMAL(10,2) DEFAULT 0;
    DECLARE descuento DECIMAL(5,2) DEFAULT 0;
    DECLARE total DECIMAL(10,2) DEFAULT 0;
    DECLARE idCurso INT DEFAULT 0;
    DECLARE idDescuento INT DEFAULT 0;

    SELECT IFNULL(pc.costo, 0), IFNULL(pc.id_curso, 0)
    INTO costo, idCurso
    FROM PERIODO_CURSO pc
    WHERE pc.id_periodo_curso = NEW.id_periodo_curso
    LIMIT 1;

    SELECT d.id_descuento, IFNULL(d.porcentaje_descuento, 0)
    INTO idDescuento, descuento
    FROM RECOMPENSA_CANJEADA rc
    JOIN DESCUENTO d ON rc.recompensa = d.id_descuento
    WHERE rc.id_estudiante = NEW.id_estudiante
    AND d.id_periodo_curso = NEW.id_periodo_curso
    ORDER BY rc.fecha_recompensa DESC, rc.hora_recompensa DESC
    LIMIT 1;

    IF idDescuento IS NULL THEN
        SET idDescuento = 0;
        SET descuento = 0;
    END IF;

    INSERT INTO BITACORA_INSCRIPCION_ESTUDIANTE (accion, fecha, descripcion)
    VALUES (
        'INSERT',
        NOW(),
        CONCAT(
            'C: ', IFNULL(idCurso, 0),
            ' - PC: ', IFNULL(NEW.id_periodo_curso, 0),
            ': Est: ', IFNULL(NEW.id_estudiante, 0),
            ', Costo: ', FORMAT(IFNULL(costo, 0), 2), ' $',
            ', Descuento: ', FORMAT(IFNULL(descuento, 0), 2), '%',
            ', TOTAL: ', FORMAT(IFNULL(costo - (costo * (descuento / 100)), 0), 2), ' $'
        )
    );
END
$$
DELIMITER ;
-- Disparadores `entregas`
```

```

--
DELIMITER $$
CREATE TRIGGER `actualizar_puntos_nota` AFTER UPDATE ON `entregas` FOR EACH ROW BEGIN
    IF NEW.nota <> OLD.nota THEN
        UPDATE puntos
        SET
            saldo_actual = saldo_actual + (NEW.nota * 0.3),
            puntos_totales = puntos_totales + (NEW.nota * 0.3),
            rankingPoints = rankingPoints + NEW.nota
        WHERE id_user = NEW.id_user;
    END IF;
END
$$
DELIMITER ;
-- Disparadores `inscripcion`
--
DELIMITER $$
CREATE TRIGGER `trg_update_cupos` AFTER INSERT ON `inscripcion` FOR EACH ROW BEGIN
    UPDATE periodo_curso
    SET cupos_ocupados = cupos_ocupados + 1
    WHERE id_periodo_curso = NEW.id_periodo_curso;
END
$$
DELIMITER ;
-- Disparadores `recompensa_canjeada`
--
DELIMITER $$
CREATE TRIGGER `resta_recompensa_canjeada` AFTER INSERT ON `recompensa_canjeada` FOR EACH ROW BEGIN
    DECLARE costo DECIMAL(10,2);
    IF LEFT(NEW.recompensa, 3) = 'RC-' THEN
        SELECT costo_canje INTO costo
        FROM cosmetico
        WHERE id_cosmetico = NEW.recompensa;
    ELSEIF LEFT(NEW.recompensa, 3) = 'RD-' THEN
        SELECT costo_canje INTO costo
        FROM descuento
        WHERE id_descuento = NEW.recompensa;
    END IF;
    IF costo IS NOT NULL THEN
        UPDATE puntos
        SET
            saldo_actual = saldo_actual - costo,
            puntos_gastados = puntos_gastados + costo
        WHERE id_user = NEW.id_estudiante;
    END IF;
END
$$
DELIMITER ;

```