

Proyecto semestral parte 0 - PHP-linux

Entrega

Fecha de entrega: domingo 25 agosto 23:59, informe por Canvas, código y datos en el server
Consultas por issues de GitHub hasta el viernes 23 a las 18:00
Modalidad: Trabajo Individual

Objetivo General del proyecto semestral

El objetivo general del proyecto semestral es que el estudiante aprenda, en la práctica, las tecnologías utilizadas para trabajar con bases de datos relacionales mediante el uso de tecnologías PHP, Web y Postgresql en un ambiente cliente-servidor real. El segundo objetivo es que el estudiante use los conceptos vistos en clase para modelar e implementar una solución para un problema real.

Objetivo Específico Etapa 0

El objetivo específico de esta entrega es que el estudiante aprenda y se **familiarice con el lenguaje PHP**, el **manejo básico en el sistema operativo Linux** y el **procesamiento y limpieza de datos** almacenados en archivos. En esta etapa, cada estudiante en forma individual, mediante la construcción de una aplicación simple en PHP, deberá:

- Cargar en su cuenta del equipo Linux los archivos entregados en el repositorio (ftp)
- Leer los datos desde los archivos entregados
- Limpiar los datos que no sean compatibles con las reglas de negocio expuestas más adelante
- Cargar de los datos limpios en tablas implementadas como Matrices
- Procesar algunas consultas sobre los datos y entregar el resultado por consola o en un archivo
- Además deberá entregar un informe con la solución y la forma de ejecutar el programa.

El Proyecto Global

La Universidad "La Mejor" tiene un sistema de registro académico de los años 80 del siglo pasado, que corre solamente en servidores mainframe antiguos y con clientes en Windows 95 con Explorer 7. La Facultad de "Humanidades y Filosofía" necesita, para su gestión administrativa académica, un mejor sistema de registro y ha decidido hacer un cisma.

En un acto de creatividad, Ud. y su grupo ofrece a la facultad construir la aplicación *Bananer*, cuyo objetivo es llevar el registro de Programas, estudiantes, cursos, notas, etc. Lamentablemente, no es posible integrar el sistema de registro actual con nada mediante servicios, solo a

través de archivos de reporte en texto extraído del sistema actual.

La aplicación *Bananer*, cuando esté completa (a final de semestre), deberá entregar una gran experiencia de servicios a los estudiantes, profesores y administrativos de la facultad facilitándoles de manera extraordinaria el proceso de ingreso de notas finales, seguimiento académico, inscripción de ramos, reportería estándar y acceso a reportería personalizada.

Requerimientos y descripción para la etapa 0: "Preparación de la migración"

Como preparación a la construcción de *Bananer* se va a realizar una prueba de carga de datos la que servirá como ejercicio de preparación para la Etapa 2.

En cada migración de sistemas (en este caso el Legacy a Bananer) se encontrarán con datos inconsistentes, mal formateados o faltantes así que en esta etapa Ud. a modo de ejercicio, deberá lidiar con estos problemas en PHP, para luego en la Etapa 2 lo haga usando Bases de Datos. Para esta etapa 0 Ud. (el trabajo es individual) deberá:

- Crear una aplicación Bananer.php en PHP, ejecutable por consola que lea los datos desde los archivos CSV entregados.
- Realice la detección de los datos que no cumplan con los formatos correctos, indicados más adelante y si es posible los corrija, de lo contrario lo descarte.
- Cargue los datos limpios en las tablas, implementadas como matrices de PHP, indicadas más adelante.
- Realizar las siguientes consultas sobre los datos, entregando el resultado por consola o en archivo:
 1. Carga académica acumulada: Entregar, en una línea, cada curso realizado (cualquiera sea la calificación) por un estudiante identificado por su **RUN ingresado por consola**, indicando: periodo del curso, sigla del curso, curso, nota y calificación. Al final de cada período la nota promedio del semestre (PPS) y al final del reporte la nota promedio de todos los cursos (PPA)
 2. Lista de curso: Entregar, en una línea, cada estudiante perteneciente a un curso y periodo, especificado por consola, indicando: cohorte, nombre completo, RUN y número de estudiante

Reglas de negocio

Las reglas de negocio es la especificación o restricción que define cómo se debe realizar una operación dentro de una organización. Las reglas de negocio para este problema son:

- Un archivo contiene información de los **cursos** que se han impartido desde el comienzo de los tiempos. El contenido específico está más adelante.
- El segundo archivo contiene la información de los profesores y administrativos vigentes.
- Los profesores y administrativos se identifican por RUN
- Los estudiantes se identifican por RUN o número de estudiante. Un RUN puede tener más de un número de estudiante, cada vez que la persona entra a primer año en la universidad se le asigna un nuevo número.
- Los estudiantes pueden hacer cambio de plan sin que ello implique un cambio de número de estudiante.
- Un estudiante puede suspender estudios y al volver se le cambia la cohorte del periodo de retorno.

- Las notas van de 1.0 a 7.0 o vacío. Vacío se usa para los cursos que aún no terminan o los cursos convalidados o eximidos.
- Todos los planes duran 10 semestres, cada semestre se llama nivel, los niveles son: Ingreso, 1...10, licenciado. Ingreso corresponde a los estudiantes ingresados en el período actual (período que no ha terminado), Licenciado corresponde a los que además de cumplir exitosamente los 10 semestres, aprobaron el examen de grado.

El contenido de los archivos de datos es el siguiente:

Archivo 1

- Cohorte: string, Período de ingreso o re-ingreso del estudiante, no nulo
- Código Plan: string, Código del plan de estudios, no nulo
- Plan: string, Nombre del plan de estudios, no nulo
- Bloqueo: Boolean, Indica si el estudiante tiene alguna restricción (deuda), no nulo
- RUN del estudiante: int, Rol Único Nacional, no nulo
- DV del estudiante: char, Dígito verificador del RUN, no nulo
- Nombres: string, todos los nombres propios, no nulo
- Apellido Paterno: string, Apellido Paterno, no nulo
- Apellido Materno: string, Apellido Materno, admite nulos
- Nombre completo: string, Nombres + Apellido Paterno+Apellido Materno, no nulo
- Número de estudiante: int, identifica en forma única a un estudiante y su plan de estudio, no nulo
- mail personal: string, Correo electrónico personal, admite nulos
- mail institucional: string, Correo electrónico institucional @lamejor.cl, admite nulos
- Periodo curso: string, periodo en que se cursó o está cursando la asignatura, no nulo
- sigla curso: string, código que identifica a una asignatura, no nulo
- curso: string nombre de la asignatura, no nulo
- Sección: sección de la asignatura, no nulo
- Nivel del curso: semestre al que pertenece la asignatura en la malla, no nulo
- Calificación: string, resultado conceptual de la evaluación obtenida, no nulo
- Nota: float 1.0 a 7.0 o vacío, resultado numérico de la evaluación obtenida, puede contener nulo
- Último logro: Nivel del curso más atrasado aprobado por el estudiante, no nulo
- Fecha Logro: string, período en que se obtuvo el Último logro, no nulo
- Última toma de ramos: Período de la última inscripción de asignaturas, no nulo

Archivo 2

- RUN del profesor/administrativo: int, Rol Único Nacional, no nulo
- DV del profesor/administrativo: char, Dígito verificador del RUN, no nulo

- mail personal: string, Correo electrónico personal, admite nulos
- mail institucional: string, Correo electrónico institucional @lamejor.cl, admite nulos
- teléfono: int de 9 dígitos, teléfono personal, admite nulos
- contrato: string, tipo de contrato (fuul, part-time, hora, etc.), admite nulos
- jornada diurna: string, "DIURNO", admite nulos
- jornada vespertina: string, "vespertino", admite nulos
- dedicación: string, horas del contrato, admite nulos
- grado académico:string, grado académico "Bachiller", "Licenciado", "Doctor", admite nulos
- jerarquia: String, jerarquía académica, admite nulos
- cargo: string, cargo administrativo, admite nulos

Y las matrices a implementar son:

1. Persona: Datos comunes de la persona (sin registros duplicados)

- RUN
- DV
- Nombres
- Apellido Paterno
- Apellido Materno
- Nombre completo
- Teléfono
- mail personal: string, Correo electrónico personal
- mail institucional: string, Correo electrónico institucional @lamejor.cl

2. Estudiante: Datos del estudiante (sin registros duplicados)

- Cohorte
- Número de estudiante
- Código Plan
- último Logro
- Fecha Logro
- Última toma de ramos

3. Profesor: Datos de Profesor

- RUN
- DV
- contrato: string, tipo de contrato (fuul, part-time, hora, etc.), admite nulos
- Jornada: Diurno o Vespertino
- dedicación: string, horas del contrato, admite nulos
- grado académico:string, grado académico "Bachiller", "Licenciado", "Doctor", admite nulos

- jerarquia: String, jerarquía académica, admite nulos
- cargo: string, cargo administrativo, admite nulos

4. Administrativo: Datos de Administrativo

- RUN
- DV
- cargo: Cargo en la universidad, admite nulos

5. Cursos: Stock de cursos y datos asociados

- sigla curso, no nulo
- curso, no nulo
- Secciones: int, número de secciones del curso, no nulo
- Nivel del curso, no nulo

6. Notas

- Número de estudiante, no nulo
- sigla curso, no nulo
- Periodo curso, no nulo
- Calificación, no nulo
- Nota, admite nulos

Nota 1: Si lo encuentra necesario, puede definir otras matrices, además de los indicados, más arriba y entregar el archivo con su contenido. En este caso, Ud. deberá dejarlo plasmado en el informe.

Nota 2: No nulo significa que el campo debe tener un valor válido dentro del dominio definido (string, int, boolean, etc.). Acepta nulos significa que es válido que el valor no esté definido (equivalente a vacío)

Nota 3: Ayuda para limpieza de datos:

- Si encuentra un error de tipo codificación (Por ejemplo caracter tilde no soportada "%'a" reemplácelo por el correcto "á" en el programa PHP.
- Si encuentra un valor nulo y la especificación indica no nulo, reemplácelo por una marca de valor inválido (por ejemplo en calificación, "" por "x" en el programa PHP.
- Las tablas (matrices) no admiten valores repetidos.

Temas administrativos

- Las entregas 0 a 3 son Incrementales, cada una puede agregar, quitar o cambiar datos, reglas de negocio u otros parámetros según se vaya avanzando en la materia del curso.
- El Código PHP y Datos se debe entregar en su cuenta en el servidor linux y el informe por Canvas
- El contenido del informe debe contemplar:
 1. Análisis de los datos (tipos) entregados en los archivos
 2. Tipos de errores de datos detectados por el programa y forma de solución utilizada.
 3. Nombre de los archivos de salida y explicación de su contenido
 4. Instrucciones para ejecutar el programa (muy importante)