Entrega 3: Integración de datos de distintas fuentes

1. Descripción General

El objetivo de esta entrega es unir los datos de un grupo par y un grupo impar, para armar un sistema integrado de vuelos, y usar SQL avanzado y procedimientos almacenados para ayudar a la gestión automática de datos.

En esta fase aprenderán a implementar una aplicación que requiere consumir dos bases de datos distintas. También conocerán de primera mano los problemas de manejo de datos que ocurren al integrar bases de datos en el mundo real y los problemas al recibir bases de datos creadas por otra persona. Los grupos deberán afrontar una serie de desafíos ligados a estos problemas.

2. Detalles Académicos

Se deberá juntar un grupo **par** con uno **impar**. Los grupos que deseen trabajar juntos podrán inscribirse en este **formulario** hasta el día 2 de junio, de lo contrario serán asignadas al azar. Las parejas de grupos definitivas serán publicadas el día **viernes 3 de junio**.

Ubicación de su entrega

1. La nueva versión de la app debe estar ubicada en el directorio **/home/grupoXX/Sites/**. Esto significa que el homepage debe ser **/home/grupoXX/Sites/index.php**. Usar cualquier otra ruta arriesga que el proyecto no sea corregido.

2. El procedimiento almacenado debe estar (o debe haber una copia) en la carpeta **/home/grupoXX/Entrega3/**

donde XX es el número del **MENOR** de los dos grupos unidos. Se asumirá también que la página principal de su sitio esta está ubicada en

`codd.ing.puc.cl/~grupoXX/index.php`

Importante: Esta entrega requiere una coordinación entre los grupos que no siempre es sencilla. Se recomienda planificar con anterioridad y no dejar todo el trabajo para la última semana.

3. Requerimientos

Su aplicación va a tener que integrar la base de datos de la plataforma de gestión comercial de vuelos (grupo impar), con la plataforma de DGAC (grupo par). Para trabajar en esta entrega **no se permite** migrar todo el esquema a la base de datos de uno de los grupos, sino que se deben manejar cada una en el servidor respectivo. La idea de esto es mantener dos conexiones abiertas.

Su página Web tiene que soportar las siguientes acciones y requerimientos. Lógicamente, todo lo realizado vía Web deberá verse reflejado en sus bases de datos.

3.1. Usuarios y Acceso

Sistema de usuarios: Su página debe manejar un sistema de usuarios, para lo que deben crear una tabla llamada `Usuarios` con esta información en el servidor del grupo **IMPAR**. Cada usuario tendra un id, un username, y una contraseña, y puede ser de distintos tipos, estos son:

- *Admin DGAC*: que podrá aprobar o rechazar propuestas de vuelos.
- *Compañía aérea*: que puede armar una propuesta de vuelos, o ver sus vuelos aprobados.
- *Pasajero*: que puede ver sus reservas, o generar una reserva nueva.

Su aplicación necesita tener un botón designado `importar usuarios` que hará lo siguiente:

- Creará (si es que no existe) un usuario llamado `DGAC` con la contraseña `admin` y de tipo Admin DGAC.
- Conviertirá en usuarios de tipo compañía aérea, a todas las compañías presentes en la base de datos del grupo impar (si ya no existen en la tabla `Usuarios`). El username debería ser el código de la compañía, y le deberían asignar una contraseña al azar.
- Convertirá a todos los pasajeros de la base de datos de grupo impar, a usuarios de tipo pasajero (si ya no existen en la tabla `Usuarios`). Aquí el username será el número de pasaporte, y se deberían asignar una contraseña derivada del nombre del pasajero y su número de pasaporte(deben añadir un grado de aleatoriedad en la derivación).

Importante: Esta funcionalidad se debe crear usando php o procedimientos almacenados (o ambos). La asignación de contraseñas no puede hacerse manualmente. Deben reportar el método con el que hicieron el import y las contraseñas actuales de **todos los usuarios** en el informe o en un readme para que lo podamos revisar.

Información personal y *Login*: La página de login tendrá dos campos, username y contraseña, dónde los usuarios pueden autentificarse en el sistema. Al hacer el login, y dependiendo del tipo de usuario, se permitirán distintas funciones especificadas en sección 3.2.

En resumen, su página inicial tendrá dos funcionalidades:

1. El botón para importar los usuarios;
2. Dos campos para que un usuario existente pueda hacer el login (probablemente necesitarán un botón extra para realizar el login mismo). Al realizar login con éxito, un usuario tendrá distintas opciones dependiendo de tipo de usuario.

3.2. Navegación

Dependiendo del tipo de usuario que realizó el login, se debe desplegar una página distinta, dependiendo de tipo de usuario.

Admin DGAC: Este usuario puede aprobar o rechazar las propuestas de vuelo. Al hacer el login, este usuario debería ver un listado de todas las propuestas de vuelo pendientes. El usuario puede hacer click en cada propuesta, y acceptar/rechazar a esta propuesta. Esta selección se debe reflejar en la base de datos de grupo par, y del grupo impar.

Adicionalmente, el usuario debe poder filtrar propuestas por fecha. Para esto, deben tener dos campos dónde el usuario ingresa fechas, y un botón que, al hacer click, despliega solo las propuestas en este rango de fechas, con las mismas funcionalidades descritas anteriormente.

Compañía aérea: Estos usuarios debería ver desplegado el nombre de la compañía, junto con dos listas: una con vuelos aprobados, y la otra con los vuelos rechazados.

Pasajeros: Para este tipo de usuario deben visualizar su nombre, número de pasaporte, y un listado de todas sus reservas.

Adicionalmente, debe existir una sección dónde el usuario puede hacer una reserva. En esta debe tener una vista que posea 3 campos y un botón que permita hacer una búsqueda de vuelos en base a sus caracterísiticas, estos campos serían los siguientes:

- Ciudad de origen: esto debe ser un dropdown menú con todos los posibles origenes presentes en vuelos aprobados en la base de datos del grupo par. Hint: pueden cargar esta información al cargar la página, guardarla en una variable, y listarla desde allá.
- Ciudad de destino: con las mismas restricciones cómo en el caso de ciudades de origen.
- Fecha de despegue:

Al presionar el botón, se debe desplegar la lista de vuelos aprobados para esta fecha entre estas ciudades.

Ahora, el usuario puede hacer click a un vuelo, y verá tres campos de texto en los que puede ingresar pasajeros en base a su pasaporte a quienes les quiere comprar tickets, y un botón para generar la reserva.

Al seleccionar el botón se debe verificar que haya por lo menos un número de pasaporte ingresado en alguno de los campos anteriormente mencionados, además se debe verificar internamente para cada uno de ellos que este sea válido y que el usuario en cuestión no posea un vuelo que tope **temporalmente** con el vuelo seleccionado; Además se debe corroboara que al añadir a estos usuarios al vuelo no se supera la capacidad máxima de pasajeros que puede llevar el avión. En el caso de que los datos ingresados cumplan con lo solicitado se debe generar la reserva, y actualizar las tablas en el servidor de grupo impar. En el caso contrario se debe mostrar un mensaje de error que indique claramente el motivo por el cual no se puede realizar la reserva solicitada.

Esta funcionalidad (generar la reserva) deben hacer ocupando procedimientos almacenados. Quiere decir, al hacer el click al botón para generar la reserva, se ejecuta un procedimiento en PL/pgSQL, para actualizar las tablas y hacer las verificaciones de datos descritas anteriormente.

3.3. Procedimiento Almacenado

Importante: la funcionalidad de generar la reserva (el último botón descrito para la página de pasajeros) debe ser ejecutada por medio de un procedimiento almacenado en lenguaje **PL/pgSQL**: en php se debe invocar a ese procedimiento con los datos de entrada de la consulta (además de otros necesarios, como los detalles de vuelo), y el procedimiento debe ejecutar y insertar las tuplas a las tablas de su base de datos. Adicionalmente, deberían desplegar la información de la reserva qué se genera.

3.4. Creatividad

Funcionalidad Adicional: Debe agregar una funcionalidad adicional que no esté listada en el enunciado y que aporte valor a la aplicación. Por ejemplo, no se podría hacer una vista de pasajes de un pasajero la cual posea filtros de tiempo, origen y destino o podrían añadir una funcionalidad que permita a la aerolinea crear propuestas de vuelo que se reflejen en ambas bases de datos u otro.

4. Bonus

No se evaluará la calidad de la página, ni su atractivo visual; sin embargo, los ayudantes tendrán la facultad de ofrecer un bonus de 5 décimas extra en la nota de esta entrega a

aquellos grupos que presenten una página con una navegación sobresaliente. Deberá notarse un trabajo adicional al de la entrega anterior.

5. Entrega

El plazo para esta entrega es el **Viernes 24 de Junio a las 23:59 horas**.

Debe entregar los siguientes items:

- **Aplicación Web**: Todos los archivos necesarios para el correcto funcionamiento de la aplicación en su carpeta Sites.
- **Readme.md o Reporte.pdf**: Donde indique cómo logearse en la aplicación (contraseñas), cómo corregir la funcionalidad adicional que implementaron y cómo corregir cada requerimiento de esta entrega si no es intuitivo desde la *homepage*. Adicionalmente, deberían especificar cómo asignaron contraseñas a los usuarios existentes, y cómo registraron al personal administrativo cómo usuarios. Los ayudantes se reservan el derecho a descontar décimas en una entrega en la que se haya dificultado la corrección. Archivo ubicado en la carpeta Entrega3 (afuera de Sites).
- **Procedimiento Almacenado**: Archivo(s) con la definición de su procedimiento almacenado, en lenguaje **PL/pgSQL**. Ubicado junto al Readme en la carpeta Entrega3.

6. Evaluación de pares

Cómo aquí trabajarán en grupos, también existe una evaluación de pares. Esto puede afectar a su nota (y disminuirla) si sus pares encuentran qué no aportaron al proyecto. Para aprobar el proyecto, deben contar con una nota promedio de al menos 4 (de 7) entre todos sus pares.