

## **Desarrollo de Sistemas de Información (DESI) – 2023**

### **TRABAJO PRÁCTICO INTEGRADOR – ACTIVIDAD 02**

#### **Descripción**

Dado el enunciado del TP Integrador y las historias de usuario presentadas a continuación, se le solicita desarrollar un sistema de información que atienda las necesidades planteadas por el usuario.

#### **Tareas de preparación**

- Para comenzar, deberá re-leer detallada y cuidadosamente el enunciado general del TP Integrador.
- Todas las dudas y consultas que genere dicha lectura las deberá consultar en el foro del tp integrador. Tenga en cuenta que usted representa, en esta situación, al profesional de software que deberá asesorar y trabajar en el problema planteado por su cliente, en este caso, el docente a cargo de la materia.

#### **Tareas a realizar**

Cada grupo deberá crear “un único” repositorio en github.com que utilizaran para compartir avances entre sus integrantes y trabajar en equipo. Cada integrante del equipo descargará de forma local el repositorio y subirá allí su trabajo, para lo cual deberá contar con una cuenta en github.

Cada equipo tendrá como objetivo desarrollar una aplicación web Java utilizando las herramientas, frameworks y conceptos discutidos en clase.

A continuación se presenta el product backlog del cual cada integrante del equipo tomará un requerimiento a implementar. Es decir que un grupo de 4 integrantes deberá implementar cuatro historias de usuario mientras que uno de 3 integrantes deberá implementar 3 historias. El objetivo es que, si bien se lleve adelante un trabajo en equipo, lo más colaborativo posible, cada historia de usuario será “tomada” por un integrante diferente, quien será responsable de su implementación.

**Como** Operador de aerolínea

**Quiero** poder programar un vuelo

**Para** que los clientes y vendedores cuenten con dicho vuelo a la hora de adquirir un ticket.

*Estimación: 8    Prioridad: **Muy Alta**    Riesgo: **Bajo***

Criterios de Aceptación:

- ✓ El usuario deberá cargar los siguientes datos básicos del vuelo :
  - Nro de vuelo ( valor alfanumérico)
  - Origen (lista de selección con todas las ciudades registradas en el sistema)
  - Destino (lista de selección con todas las ciudades registradas en el sistema)
  - Tipo de vuelo (nacional/internacional)
  - Precio del pasaje (Nro positivo, con dos decimales. Aclaración: por simplicidad habrá un solo precio para todos los asientos)
  - Fecha y hora de partida
  - Avión (lista seleccionable de aeronaves registradas en el sistema).  
Tenga presente que cada avión tendrá información de su capacidad (cant filas y nro de asientos por fila ) y ésta determina la capacidad del vuelo
- ✓ No podrá haber dos vuelos para el mismo día, para un mismo avión
- ✓ El nro de vuelo debe ser único
- ✓ Todos los datos son requeridos
- ✓ Todos los vuelos se registran con el estado "Normal"

**Como** Operador del aeropuerto

**Quiero** poder editar datos impositivos y de tasas

**Para** que puedan ser utilizados a la hora de calcular el importe a abonar por un pasaje aéreo

Estimación: 8    Prioridad: **Alta**    Riesgo: **Bajo**

Criterios de Aceptación:

- ✓ El sistema deberá permitir la edición de todas las tasas e impuestos necesarios para calcular el importe de un pasaje:
  - Iva (expresado en %, min:0, max:100)
  - Tasa aeroportuaria nacional (importe expresado en \$)
  - Tasa aeroportuaria internacional (importe expresado en USD)
  - Cotización del dólar (importe expresado en \$)
- ✓ Todos los datos son requeridos
- ✓ Todos los valores son numéricos positivos, con dos decimales
- ✓ No se permite agregar ninguna nueva tasa un eliminar una existente.

**Como** Operador de aerolínea / Cliente

**Quiero** poder listar los vuelos programados para el día

**Para** poder informar a los pasajeros acerca de próximos vuelos, horarios y estado del vuelo

Estimación: 8    Prioridad: **Muy Alta**    Riesgo: **Bajo**

Criterios de Aceptación:

- ✓ El usuario ingresa:
  - Fecha a consultar (filtro obligatorio)
  - Origen (filtro opcional, lista de selección con todas las ciudades registradas en el sistema)
  - Destino (filtro opcional, lista de selección con todas las ciudades registradas en el sistema)
  - Tipo de vuelo (filtro opcional, valores posibles: nacional/internacional)
- ✓ La lista de vuelos mostrará las siguientes columnas:
  - Fecha
  - Nro de vuelo
  - Tipo de Vuelo
  - Estado (Normal/Reprogramado/Cancelado)
  - Cantidad de asientos libres
- ✓ La lista se mostrará ordenada por fecha/hora de vuelo, mostrando primero las fechas mas cercanas

**Como** Operador de una agencia de turismo

**Quiero** poder vender un pasaje aéreo a un cliente registrado

**Para** que los clientes que se acercan al local, puedan adquirir su ticket de vuelo

*Estimación: 8    Prioridad: **Muy Alta**    Riesgo: **Bajo***

#### Criterios de Aceptación:

- ✓ El usuario podrá tipear un dni y el sistema verifica que el mismo se encuentre registrado en el sistema .
  - Si el cliente se encuentra registrado, se muestran sus datos básicos:
    - nombre y apellido
    - domicilio
    - e-mail
    - fecha de nacimiento
    - nro pasaporte
  - Si el cliente no se encuentra registrado, se muestra el error al usuario
- ✓ El usuario elige:
  - Vuelo (lista desplegable de selección única). Una vez elegido el vuelo, el sistema mostrará el precio a pagar (por simplicidad considere que todos los asientos cuestan el mismo importe)
  - Asiento elegido (lista desplegable que mostrará solo aquellos asientos disponibles.
- ✓ Todos los datos son requeridos
- ✓ El usuario presiona emitir ticket y el sistema muestra un mensaje indicando si el pasaje pudo ser vendido o no.

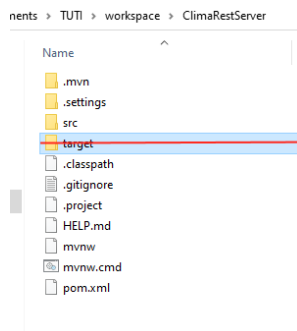
Notar que esta historia de usuario no abarca el proceso de cobro

## Formato de la entrega

Deberá generar y entregar un archivo comprimido conteniendo

- Nombres y dirección de mail de cada integrante del equipo (puede ser en un archivo txt o en el documento Word detallado en el ultimo punto)
- Datos para la descarga del repositorio creado en GitHub
- El código fuente completo del proyecto (Notar que aparte del repositorio GitHub, se pide aquí adjuntar los fuentes para que queden registrados en el campus )
- Un script para inicializar la base de datos con todos aquellos datos necesarios para poder probar el desarrollo
- Un documento Word conteniendo toda aquella aclaración/instructivo/documentación que considere necesaria para las pruebas del software entregado.

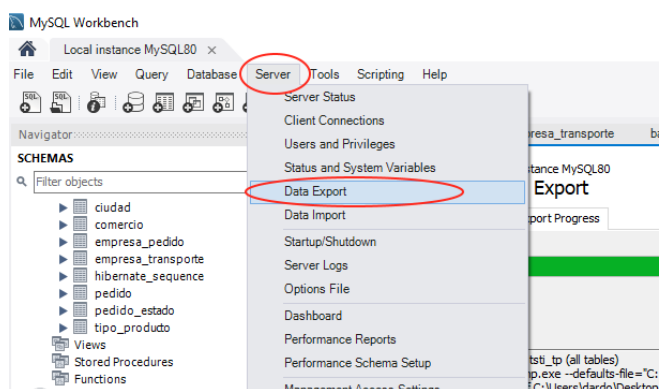
Dentro del archivo comprimido deberá incluir el proyecto Eclipse completo, con todas sus clases, archivos pom y demás archivos necesarios para correr el proyecto mediante Eclipse STS. Solo se sugiere comprimir el proyecto completo, eliminando únicamente la carpeta “target” (para reducir el tamaño del archivo generado), dado que esa carpeta se regenera automáticamente al compilar el proyecto nuevamente.



Incluya todos los scripts necesarios para poblar la base de datos (ej. para la tabla de ciudades) de modo que sus casos de prueba funcionen.

Puede generar los mismos exportando datos desde su base de datos de la siguiente manera:

## 1- Abra la opción “Data Export”



## 2- Exporte solo datos (ya que la estructura será generada automáticamente mediante el mapeador):

