

**Objetivos:** Desarrollar en los estudiantes las habilidades básicas de la programación orientada a objetos, aplicando los conceptos de excepciones, enumerados e interfaz gráfica de usuario.

## **Sistema: “SISTEMA DE SOLICITUD DE CITAS” - Entrega 3**

Se quiere construir una aplicación que ayude a los usuarios de un sistema de embajadas a solicitar citas para diferentes tipos de visa.

El sistema de citas de la embajada maneja dos listas: las visas y las solicitudes

El sistema de citas de la embajada maneja además dos maps: el de usuarios y el de requisitos de cada tipo de visa.

La aplicación debe permitir a los usuarios hacer la solicitud de la cita para una Visa.

El valor de la solicitud de visa depende de su tipo: (Turismo, Trabajo, Estudiante) y del tipo de solicitante (Niño0-2, Niño2-12, Adulto, AdultoMayor). El tipo de solicitante depende de su edad.

Por simplicidad, este sistema no le brinda una respuesta de aceptación o rechazo de su solicitud. Sólo le permite al usuario solicitar la cita, calcular el valor de una solicitud de visa y generar reportes: de las citas de una determinada fecha y de beneficiarios de exención de pago.

Para realizar una solicitud, es requisito indispensable que el usuario ya esté registrado en el sistema.

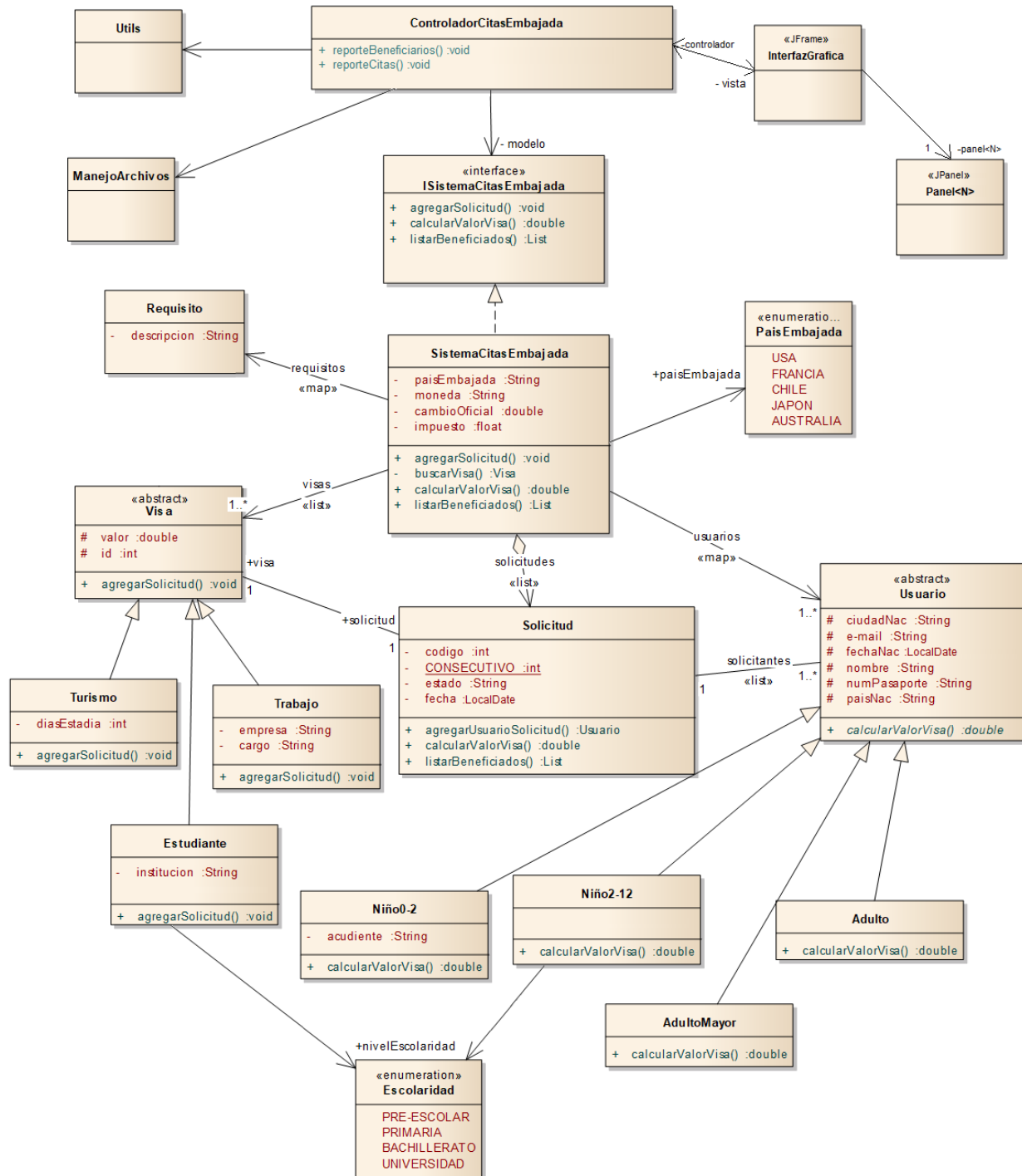
Los siguientes pasos se aplican para realizar una solicitud:

- El sistema muestra los tipos de visa que maneja (“Turismo”, “Trabajo” o “Estudiante”).
- El usuario selecciona el tipo de visa elegido.
- El sistema solicita al usuario los datos de la solicitud:
  - I. si escogió como tipo de visa una diferente a “Turismo”, el sistema pide el número de pasaporte del solicitante. El sistema busca dicho usuario y se instancia un objeto Solicitud (asignando como identificador un número consecutivo, a partir de un atributo de clase), asociando el usuario a dicha solicitud.
  - II. Si escogió como tipo de visa “Turismo”, se solicita el nombre del archivo con los números de pasaportes de los solicitantes. El sistema busca dichos usuarios y se instancia un objeto Solicitud (asignando como identificador un número consecutivo, a partir de un atributo de clase), asociando los usuarios a dicha solicitud. También se solicita el número de días de la Estadía.
  - III. Si escogió como tipo de visa "Trabajo", se le solicita el cargo y la empresa donde va a trabajar.

IV. Si escogió como tipo de visa "Estudiante" se solicita la escolaridad y la institución donde va a estudiar.

- Como se dijo antes, con estos datos, el sistema instancia un objeto Solicitud (asignando como identificador un número consecutivo, a partir de un atributo de clase y asignándole una fecha para la cita). También se instancia un objeto de una subclase de Visa asociado a la Solicitud.
- El nuevo objeto Solicitud debe ser agregado a la lista de solicitudes del Sistema. Por defecto, el sistema le asigna como estado "Pendiente".

## Diagrama de clases



## Notas del diagrama de clases

- No se muestran los detalles de los métodos: ni tipos de parámetros ni tipo de resultado.

- En el diagrama no se muestran las clases *Comparator* necesarias para elaborar reportes ordenados.
- Se agregaron las enumeraciones *PaisEmbajada* y *Escolaridad*. Las clases *Estudiante* y *Niño2-12* tienen atributos de tipo *Escolaridad* y la clase *SistemaCitasEmbajada* tiene un atributo de tipo *PaisEmbajada*.
- El nivel de Presentación está constituido principalmente por la clase *InterfazGrafica* que es un *JFrame* (ventana gráfica) que interactúa con el usuario, a través de los paneles (*JPanel*) correspondientes (En el diagrama *Panel<N>*, donde N corresponde a cada una de las pestañas del tarjetero explicado más adelante). Una vez el usuario genera un evento, el panel correspondiente recibe el evento y lo reenvía a la ventana gráfica, y ésta al controlador para que se realicen los llamados correspondientes en *ISistemaCitasEmbajada*. Una vez éste último retorne la respuesta, el controlador es el encargado en realizar el llamado a la interfaz gráfica para que actualice el panel correspondiente.
- Las clases *InterfazGrafica* y *Panel<N>* deben pertenecer al paquete *co.edu.javeriana.citasembajada.presentacion*, la clase *ManejoArchivos* al paquete *co.edu.javeriana.citasembajada.persistencia* y las demás clases al paquete *co.edu.javeriana.citasembajada.modelo*. La clase **Utils** contiene métodos utilitarios como calcular la fecha del día siguiente, la edad de un usuario, *etc.* La clase **ManejoArchivos** contiene los métodos para leer los archivos de datos.
- Se deben leer datos en la presentación (Vista) y procesarlos en la lógica de negocio (Modelo)
- Toda creación y procesamiento de objetos debe realizarse en la lógica de negocio pasando los parámetros necesarios desde la GUI (presentación)
- La interfaz **ISistemaCitasEmbajada** es el punto de comunicación entre el paquete de presentación y el de lógica de negocio. Esta interfaz enumera los servicios de negocio que ofrece la embajada a los usuarios y es el punto de entrada al nivel funcional del sistema.
- La clase **SistemaCitasEmbajada** implementa la interfaz *ISistemaCitasEmbajada*. Un objeto de esta clase tiene como atributos el país donde se encuentra la Embajada, la moneda local del país donde se encuentra la embajada, el cambio Oficial (entre la moneda local y la moneda del país del Sagrado Corazón), el impuesto que cobra el país, una lista de visas (expresado en porcentaje), una lista de solicitudes, un map de usuarios y un map de requisitos de cada tipo de visa.
- Para el **map de usuarios**, la llave es el número de pasaporte (representado como un String) y el valor es el objeto de la clase Usuario. Los **usuarios** se especializan en cuatro tipos: Niño de 0-2años, Niño de 2-12 Años, Adulto (entre 12 y 65 años) y Adulto Mayor (mayor de 65 años). Dependiendo de su edad, se calcula el valor de la Visa, por lo cual se recomienda escribir un método estático en la clase Utils llamado *edadEnAnios* que recibe dos (2) objetos *LocalDate* (el primero anterior al segundo) y calcula su diferencia en años. La signature de dicho método podría ser:

```
public static int edadEnAnnos (LocalDate f1, LocalDate f2);
```

- Para el **map de requisitos** la llave es un String (que se refiere a un tipo de visa) y el valor asociado es una lista de String (que son los requisitos de un determinado tipo de visa).
- Toda **Solicitud**
  - está asociada a un Usuario (o a varios si la solicitud es para una visa de Turismo) y a una única Visa.
  - A su vez toda Visa está asociada a una Solicitud.
  - A una solicitud se le asigna la fecha de la cita con la embajada.
- Para calcular el valor de una visa, se siguen las siguientes reglas:
  - Si el usuario es un Niño0-2, sólo paga el 10% de la tarifa de la visa.
  - Si el usuario es un Niño2-12:
    - i. La visa tiene un 5% de descuento por cada año que le falte para los 18 años.
    - ii. Si el tipo de Visa es de Turismo, tiene un incremento del 20%.
    - iii. Si la visa es de Estudiante, tiene un descuento del 30%
  - Si el usuario es un Adulto, paga tarifa plena.
  - Si el usuario es un AdultoMayor:
    - i. La visa tiene un 25% de descuento.
    - ii. Si el tipo de Visa es de Turismo, tiene un incremento del 10%.
- Las **visas** se especializan en tres tipos: Turismo, Trabajo y Estudiante. Toda visa tiene una tarifa básica que se toma como base para calcular el valor que debe pagar un solicitante de la visa. El atributo valor de una Visa se expresa en la moneda local (la del país donde se encuentra la Embajada).
- Todas las clases deben contar con los métodos get y set, así como los constructores.
- Modele los atributos lista con la clase ArrayList y los atributos de fechas con LocalDate. Los atributos Map se modelan con la clase HashMap

#### Archivos de datos de entrada

- Se realizará la persistencia como en la entrega 2 a través de archivos texto.
- Además, se deben ofrecer dos funcionalidades que corresponden a la serialización/deserialización del sistema: en la clase ManejoArchivos el método salvarSistema() permite guardar, a manera de objeto, el Sistema de Citas de Embajada. Debe ofrecerse además la operación inversa cargarSistema() que permite cargar directamente un objeto de tipo ISistemaCitasEmbajada.

#### Interfaz gráfica (GUI)

La interfaz gráfica (GUI) de usuario que debe implementar la clase InterfazGrafica debe ser diseñada siguiendo las siguientes pautas:

- La clase InterfazGrafica es la que contiene los diferentes paneles para la interacción con el usuario, un Panel por cada pestaña del tarjetero. Cada panel es el único encargado de leer datos provenientes del usuario y de imprimir resultados (entrada/salida). Esta clase declara e instancia una variable de tipo ISistemaCitasEmbajada. Constituye el nivel funcional de Presentación.

- Los paneles muestran botones que permiten al usuario solicitar los siguientes servicios (que se traducen en invocaciones de métodos a la variable que representa el sistema):
  - Asociar país a la Embajada
  - Ingresar solicitantes
  - Hacer solicitud de cita para visa de Turismo
  - Hacer solicitud de cita para visa de Trabajo
  - Calcular valor de visa
  - Reporte de citas para una determinada fecha
  - Consultar la lista de beneficiarios
  - Guardar el sistema en un archivo como un objeto
  - Cargar de un archivo el objeto sistema
- La clase **Controlador** contiene el método ejecutable main y un método auxiliar por cada opción del menú.
- Todos los archivos (para cargar, obtener y salvar la información) deben buscarse utilizando **JFileChooser**.

**Se pide implementar la siguiente funcionalidad utilizando la estructura propuesta:**

(entre [] se muestra el valor en puntos para la evaluación)

0. Mostrar al usuario en un tarjetero, el siguiente menú de opciones del sistema: [10]



1. Asociar país a la Embajada: [10]  
 Si el usuario escoge esta opción el sistema muestra la tarjeta "**AsociarPaisEmbajada**" con una interfaz gráfica similar a la siguiente:

**Asociar País de Embajada**

país

**Buscar archivo de Embajadas**

A partir del JComboBox, el usuario selecciona el nombre del país y a través de un JFileChooser indica el archivo texto en el que se encuentran los datos básicos de los países donde hay embajada. Luego, el sistema debe buscar en el archivo el nombre del país ingresado por el usuario y extraer del archivo la información relacionada con la moneda local, el impuesto que se debe reportar al fisco y la tasa de cambio. Esta información debe asignarla a los atributos de la clase SistemaCitasEmbajada. Un ejemplo del archivo que contiene la información descrita es el siguiente:

```
#EMBAJADAS
#id--pais-----moneda-----impuesto-----tasa de cambio
100 *USA          *dolar          *15%          *2000
110 *Francia      *euro           *25%          *2550
120 *Chile        *peso           *5%           *325
130 *Japon        *yen           *35%          *1562
140 *Australia   *dolar australiano *0%           *1850
#FIN
```

Se usa el \* para separar los campos. La convención #FIN indica que no hay más embajadas (fin de datos). La tasa de cambio es respecto a pesos

## 2. Ingresar solicitantes:

[10]

Si el usuario escoge esta opción el sistema muestra la tarjeta "**IngresarSolicitantes**" con una interfaz gráfica similar a la siguiente:

**Ingresar Solicitantes**

**Seleccionar archivo de Usuarios**

Usuarios

num Pass	Nombre	País Origen	ciudadNac	fechaNac	email	infoAd
111	Grettel Lahmann	USA	Chicago	2013-12-30	gth@yahoo.es	Yves Davoine
222	Yves Davoine	Francia	Grenoble	1995-12-30	davoine@gmail.com	

Cuando el usuario oprime el botón “Seleccionar archivo de Usuarios”, se despliega un JFileChooser en el que se pide el nombre y localización del archivo texto en el que se encuentran los datos básicos de los usuarios. Debe agregar al sistema cada usuario. No debe permitir ingresar un usuario con el mismo número de pasaporte de otro usuario ya ingresado. En caso de que no se pueda ingresar algún(os) usuario(s), se debe desplegar una caja de Diálogo (JDialog) con dicha información.

Una vez se procesa el archivo, la información se muestra en un JTable (se muestran al menos 5 usuarios, pero debe tener scroll para mostrar los demás). Vale la pena destacar que la interfaz que se muestra en la figura de arriba, es sólo un ejemplo).

Para todo usuario, el archivo contiene un tipo, el número de pasaporte, nombre del usuario solicitante, país de origen, ciudad y fecha de nacimiento y una dirección electrónica (e-mail).

Además:

- Si se trata de un niño de 0-2 años, el archivo informa adicionalmente el nombre de su acudiente
- Si se trata de un niño de 2-12 años, el archivo informa adicionalmente su nivel de escolaridad (basados en los valores de la enumeración Escolaridad: “preescolar”, “primaria”, “bachillerato”).

Dependiendo de la edad del solicitante se instancia un objeto de la clase Nino0-2 o Nino2-12 o Adulto o AdultoMayor. Dicho objeto se adiciona al map de usuarios del SistemaCitasEmbajada.

Un ejemplo del archivo que contiene la información descrita es el siguiente:

```
#SOLICITANTES
#numPass--nombre-----paisOrigen--ciudadNac-- fechaNac-----email-----infoAd
111      *Grettel Lahmann *USA      *Chicago      *2013-12-30 *gth@yahoo.es      *Yves Davoine
222      *Yves Davoine    *Francia *Grenoble    *1995-12-30 *davoine@gmail.com
333      *Gabriel Salas   *Chile   *Vina       *2002-05-21 *gsal@hotmail.com  *Bachillerato
444      *Keiko Tamura    *Japon   *Tokio      *1975-08-13 *keiko@gmail.com
555      *John Kidman     *Austria *Viena      *2008-08-13 *kidm@yahoo.com    *Primaria
#FIN
```

Se usa el \* para separar los campos. La convención #FIN indica que no hay más solicitantes (fin de datos).

### 3. Hacer solicitud de cita para visa de Turismo

[10]

Si el usuario escoge esta opción el sistema muestra la tarjeta "**SolicitarVisaTurismo**" con una interfaz gráfica similar a la siguiente:

### Hacer Solicitud de Visa de Turismo

Días Estadía

Cargar archivo de tarifas

No Solicitud

Usuarios						
num Pass	Nombre	Pais Origen	ciudadNac	fechaNac	email	infoAd
111	Grettel Lahmann	USA	Chicago	2013-12-30	gth@yahoo.es	Yves Davoine
222	Yves Davoine	Francia	Grenoble	1995-12-30	davoine@gmail.com	

Adicionar usuario

El sistema sigue los siguientes pasos:

- El usuario indica por cuántos días va a estar (es decir, su estadía en el JTextField “Días Estadía”).
- Cuando el usuario oprime el botón "Cargar archivo de tarifas":
  - se crea un objeto Solicitud y otro de la clase Turismo. Se asocia este objeto visa a la solicitud. El sistema asigna un número a dicha solicitud y se la muestra al usuario en el JTextField de “No Solicitud”.
  - Se asigna una fecha y hora para la cita. Se sugiere crear e invocar el método **asignarFechaHora** que le asigna la primera fecha y hora disponible para la cita (para implementar este método utilice un consecutivo en Utils que indique las horas que hay que sumar a la fecha del día de



hoy para obtener la próxima fecha a asignar a una solicitud). La signature del método podría ser:

```
public void asignarFechaHora (Solicitud sol);
```

- Además, se lee el archivo de tarifas (indicado por JFileChooser)
- Luego, se cargan en un JTable, todos los usuarios del sistema.
- Para asociar cada usuario a la solicitud, se escoge el usuario del JTable y se presiona el botón “Adicionar Usuario”.

La estructura del archivo de tarifas es la siguiente (el valor de la visa se expresa en la moneda del país en donde se encuentra la Embajada):

```
#VISAS
#id--tipo-----valor
10  *Turismo      *70
20  *Estudiante   *50
30  *Trabajo      *200
#FIN
```

#### 4. Hacer solicitud de cita para Visa de Trabajo

[10]

Si el usuario escoge esta opción el sistema muestra la tarjeta "**Solicitar Visa Trabajo**" con una interfaz gráfica similar a la siguiente:

Si se desea crear una solicitud de visa de Trabajo:

### Hacer Solicitud de Visa de Trabajo

Empresa

Universidad Pontificia

Cargo

Ingeniero de Sistemas

Cargar archivo de tarifas

No Solicitud

837

Usuarios

num Pass	Nombre	Pais Origen	ciudadNac	fechaNac	email	infoAd
111	Grettel Lahmann	USA	Chicago	2013-12-30	gth@yahoo.es	Yves Davoine
222	Yves Davoine	Francia	Grenoble	1995-12-30	davoine@gmail.com	

Asociar usuario

- El usuario debe indicar la empresa y el cargo (en los correspondientes JTextField).
- Cuando el usuario oprime el botón "Cargar archivo de tarifas":
  - se lee el archivo de tarifas (indicado por JFileChooser)
  - Se debe crear un objeto Solicitud y un objeto de la clase Trabajo. Para esto, Se asocia este objeto visa a la solicitud.
  - El sistema asigna un número a dicha solicitud y se la muestra al usuario en el JTextField de “No Solicitud”.
  - Se asigna una fecha y hora para la cita.
  - Luego, se deben cargar en un JTable, todos los usuarios del sistema.

- Para asociar el usuario a la solicitud, se escoge el usuario del JTable, se presiona el botón de “Asociar usuario”.

5. Calcular valor de visa:

[20]

Si el usuario escoge esta opción el sistema muestra la tarjeta "**CalcularValorVisa**" con una interfaz gráfica similar a la siguiente:

**Calcular Valor de Visa**

Búsqueda por:  

Número  

num Pass	Nombre	País Origen	ciudadNac	fechaNac	email
222	Yves Davoine	Francia	Grenoble	1995-12-30	davoine@gmail.com

num Pass	Nombre	Fecha Nac	valorVisa	impuesto	valorTotal
222	Yves Davoine	1995-12-30	150,000	37,500	187,500
111	Grettel Lahmann	2000-12-30	120,000	30,000	150,000

PSC: 862,500      Moneda Local: EURO  
Tasa de Cambio: 2550      Valor Moneda Local: 339

- Del JComboBox “Búsqueda por:”, el usuario selecciona por qué criterio buscar (número de pasaporte o código de la solicitud). De acuerdo a esta selección, el usuario ingresa en el JTextField “Número”, su número de pasaporte, o el número de solicitud
- El usuario pulsa el botón “Calcular Valor Visa” y el sistema calcula el valor total de la solicitud de la visa de acuerdo a su tipo y a la edad del (de los) solicitante(s) de la misma.
- Después de que el sistema calcula el valor de la visa, aparece un primer JTable con la información del usuario asociado a dicha solicitud o número de pasaporte. De dicho usuario aparece el número de pasaporte, nombre, país de origen, ciudad y fecha de nacimiento y, correo electrónico.
- Además, aparece un segundo JTable con la información de todos los usuarios asociados a esa solicitud con su correspondiente valor (ver figura arriba).
- Finalmente, aparecen etiquetas (labels) con el valor en la moneda del país del Sagrado Corazón, la tasa de cambio, la moneda local y la conversión a dicha moneda local.

Vale la pena indicar, que puede que los cálculos de la visa mostrados en la figura no correspondan a los valores reales.

6. Reporte de citas para una determinada fecha

[10]

Si el usuario escoge esta opción, el sistema muestra la tarjeta "**ReporteCitas**" con una interfaz gráfica similar a la siguiente:

**REPORTE DE SOLICITUDES EMBAJADA DE AUSTRIA**

Fecha:  Generar Reporte

numPass	Nombre	tipoVisa	numSolicitud
2286974	Aurelie Arnaud	Trabajo	256897
333	Gabriel Salas	Turismo	356248
111	Grettel Lahmann	Turismo	259648
212055	Isabela Mendoza	Estudiante	350214
555	John Kidman	Turismo	222111
444	Keiko Tamura	Turismo	333222
222	Yves Davoine	Turismo	123456

El usuario escribe en el JTextField “Fecha”, una fecha (en formato “aaaa-mm-dd”).

Al presionar el botón: “Generar Reporte”, aparece un JTable con la información de todos los usuarios que tienen cita ese día (como lo muestra la figura de arriba). Para cada usuario se debe mostrar su número de pasaporte, nombre completo, tipo de visa a solicitar y el número de solicitud. El reporte debe estar ordenado alfabéticamente por nombre de solicitante.

Además, se debe generar un archivo texto cuyo nombre es: **Citas-fecha.txt**. Para el ejemplo anterior, el archivo se debe llamar: **“Citas-2014-05-28.txt”**

7. Consultar la lista de beneficiarios

[10]

Si el usuario escoge esta opción, el sistema muestra la tarjeta "ReporteBeneficiarios" con una interfaz gráfica similar a la siguiente:

**REPORTE DE BENEFICIARIOS**

numPass	Nombre	Valor total
333	Gabriel Salas	187,500
111	Grettel Lahmann	150,000
555	John Kidman	150,000
444	Keiko Tamura	187,500
222	Yves Davoine	187,500

Valor total que se dejó de recaudar:

Como lo muestra la imagen, el reporte se muestra en un JTable con la información de los usuarios que son beneficiarios. El sistema genera un reporte con todos los solicitantes que no pagan su visa ya sea por ser menor de 12 años o por ser mayor de 65 años. De cada usuario sólo se desea conocer su nombre y su número de pasaporte y lo que debía haber pagado.

Al final, aparece en un JTextField, el valor total que dejó de recaudar la embajada. Este valor debe expresarse en la moneda del país del sagrado corazón (MPSC).

Vale la pena indicar que puede que en la figura los cálculos de la visa no correspondan a los valores reales.

8. Salvar los datos del sistema [5]  
El botón "**Salvar los datos del sistema**" debe permitir grabar la variable que representa todo el sistema como un objeto serializado. El sistema solicita al usuario que indique la localización y el nombre del archivo (a través de un JFileChooser). Debe haber manejo de excepciones mostrando los posibles errores. Los errores deben ser mostrados en cajas de diálogo (JDialog).
9. Cargar los datos del sistema [5]  
El botón "**Cargar los datos del sistema**" debe permitir cargar un objeto serializado contenido en el archivo indicado por el usuario: se descarta entonces la versión de la variable que representa el sistema y se le asigna el objeto cargado. El archivo del que se quiere cargar el objeto debe ser seleccionado a través de un JFileChooser. Debe haber manejo de excepciones mostrando los posibles errores. Los errores deben ser mostrados en cajas de diálogo (JDialog).

**Total de puntos: 100**

## Condiciones de la Entrega

### 1. Fecha

La entrega se hace a través de la Actividad establecida para tal fin del sistema de aprendizaje UVirtual (BlackBoard) a más tardar el **Jueves 21 de noviembre de 2019 a la hora de sustentación (9am)**

- Entregas posteriores a esta fecha no serán tenidas en cuenta.
- Si realiza varias entregas en el sistema, el grupo de trabajo, sólo se tendrá en cuenta aquella con fecha más reciente.

### 2. Grupos

La entrega se realizará en grupos de trabajo de máximo 2 personas. Los grupos no podrán cambiar su conformación y desde el comienzo dichos grupos estarán identificados plenamente.

### 3. Entregables

Un solo archivo .zip nombrado de la siguiente manera: **proy3-Nombre1Apellido1-Nombre2Apellido2.zip** en donde Nombre1Apellido1 es el nombre y apellido del primer integrante del grupo, y Nombre2Apellido2 es el nombre y apellido del segundo integrante del grupo. Dentro de este archivo comprimido deben ir los siguientes cuatro archivos:

- Archivo fuentes.zip con el código fuente de las clases (archivos .java)
- Archivo test.jar con el código ejecutable que invoca el método main de Controlador (archivos .class)
- Archivo ejecutarTest.bat que ejecuta el jar anterior
- Archivo doc.zip con la documentación javadoc del sistema

### 4. No es necesario entregar los archivos de datos (texto). La revisión se hará con archivos consistentes colocados en la raíz del proyecto.

### 5. Observaciones

- Si no se entregan los archivos enumerados previamente, no se calificará la entrega.
- Se reducirán puntos por malas prácticas de programación:
  - Código "quemado" no autorizado. Por ejemplo, usar valores constantes en donde no se deba.
  - No utilización de los criterios de asignación de responsabilidades a las clases

- No respetar la organización en paquetes
- El diagrama de clases y la implementación deben ser concordantes.
- SUSTENTACION INDIVIDUAL: en caso de no ser exitosa la sustentación, se reconocerá sólo el 30% del total de puntos obtenidos en la entrega.
- El aplicativo debe funcionar en las máquinas de la Universidad.
- Se deben cumplir las Reglas de Juego del curso.
- Para cada servicio ofrecido, en caso de que se presente una excepción, ésta debe mostrarse en una ventana emergente (por ejemplo, un JDialog)

#### **6. Restricciones**

- La GUI debe llenarse desde los objetos de negocio.
- Para las colecciones NO use arreglos []

**Sustentación: Jueves 21 de noviembre de 2019 - 9am**