**Pontificia Universidad Javeriana**

**Programación Orientada a Objetos**

**Proyecto Semestre 2018-1**

**Objetivos:** Desarrollar en los estudiantes las habilidades básicas de la programación orientada a objetos, aplicando los conceptos de herencia.

**Sistema: “SISTEMA DE SOLICITUD DE CITAS” - Entrega 2**

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 (tipos) y las solicitudes

El sistema de citas de la embajada maneja además un map: el de usuarios.

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

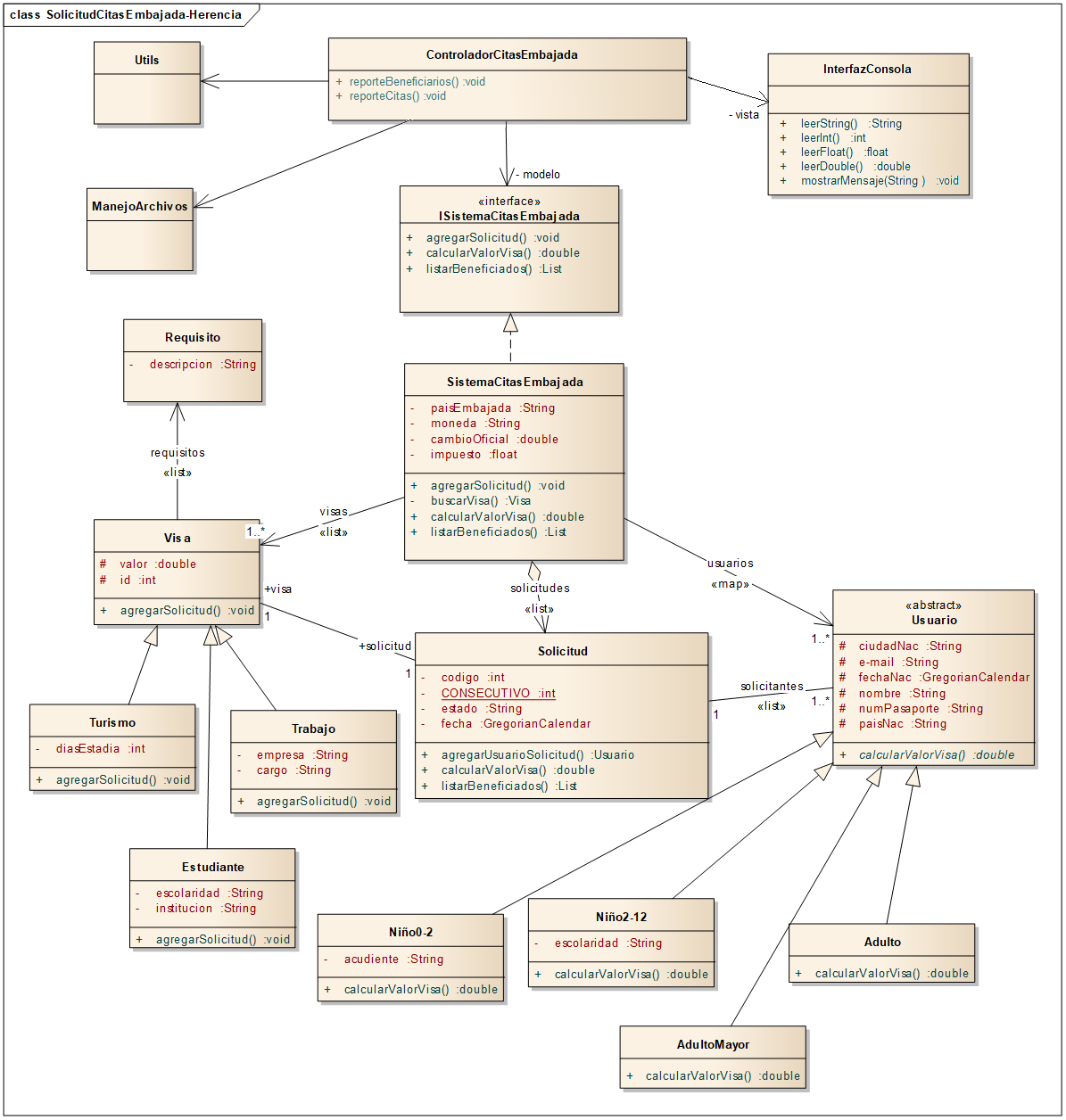
Por simplicidad, este sistema no le brinda una respuesta de aceptación o rechazo de su solicitud. Sólo le permite al usuario solicitar la cita, calcular el valor de una solicitud de visa y generar reportes: de las citas de una determinada fecha y de beneficiarios de exención de pago.

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:
  1. 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.
  2. 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.
  3. Si escogió como tipo de visa "Trabajo", se le solicita el cargo y la empresa donde va a trabajar.
  4. 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).
* 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. Usted debe identificarlos manteniendo las buenas prácticas, distribución de responsabilidades entre clases y buscando la mantenibilidad y facilidad para las modificaciones futuras.
* La interfaz ISistemaCitasEmbajada 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, una lista de solicitudes y un map de usuarios.
* 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.
* Toda solicitud está asociada a un usuario (o a varios si la solicitud es para una visa de Turismo) y a un único tipo de Visa. Por simplicidad, un usuario puede tener a la vez una sola solicitud activa y para un solo tipo de visa.
* Además, una solicitud tiene una lista de solicitantes (de tipo Usuario) asociados si es para una visa de Turismo. De lo contrario, sólo tiene un usuario asociado.
* A una solicitud se le asigna la fecha de la cita con 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
* 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.
* 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); el atributo impuesto de la clase SistemaCitasEmbajada se expresa en porcentaje**.**
* La clase “**InterfazConsola**” es el único punto en donde se puede solicitar e imprimir información (entrada y salida de datos) y sus métodos deberán ser llamados y coordinados por la clase **ControladorCitasEmbajada**.
* La clase **ControladorCitasEmbajada** es la encargada de coordinar y organizar el llamado de los métodos entre el modelo y la vista. En ningún caso deberá existir comunicación directa entre el modelo y la vista, todo se debe mediar a través del controlador.
* Para la entrada de los datos al sistema:
  1. La clase **ControladorCitasEmbajada** debe solicitar la impresión del menú de opciones a la clase **InterfazConsola**. Seguido, a través de la misma clase, debe solicitar la entrada de datos por consola. Se debe manejar un menú para proveer las diferentes opciones.
  2. Una vez la clase **ControladorCitasEmbajada** identifique cuál fue la opción seleccionada, llamará a los métodos auxiliares de dicha opción para que continúe la ejecución en el paquete modelo.
  3. Los datos son ingresados a través de archivos de texto cuyo formato se explica en cada una de las opciones. Se probará con archivos colocados en la raíz del proyecto, por lo cual debe usar en sus fuentes un objeto File inicializado así:

File f = new File("./" + nombreArchivo);

* Las clases **TestCitasEmbajada** y **Utils** deben pertenecer al paquete **co.edu.javeriana.citasembajada.presentacion**; la clase **TestCitasEmbajada** contiene el método ejecutable main y un método auxiliar por cada opción del menú; 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 y debe pertenecer al paquete **co.edu.javeriana.citasembajada.persistencia**
* Las demás clases pertenecen al paquete **co.edu.javeriana. citasembajada.modelo**

**Se pide implementar la siguiente funcionalidad utilizando el menú principal:**

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

1. Mostrar al usuario el siguiente menú de opciones del sistema, en el que la última opción debe terminar la ejecución del programa: **[5]**

SISTEMA SOLICITUD DE CITAS PARA EMBAJADA

opción 0: asociar país a la Embajada

opción 1: ingresar solicitantes

opción 2: ingresar tipos de visa y requisitos

opción 3: hacer solicitud de cita para visa de Turismo

opción 4: hacer solicitud de cita para visa diferente a Turismo

opción 5: calcular valor de visa

opción 6: reporte de citas para una determinada fecha

opción 7: consultar la lista de requisitos de un tipo de Visa

opción 8: consultar la lista de beneficiarios

opción 9: salir

1. Asociar país a la Embajada: **[10]**

Si el usuario escoge esta opción, el sistema debe pedirle el nombre del país y del archivo texto en el que se encuentran los datos básicos de los países donde hay embajada. Luego, el sistema debe buscar 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 \*dólar \*15% \*2000

110 \*Francia \*euro \*25% \*2550

120 \*Chile \*peso \*5% \*325

130 \*Japón \*yen \*35% \*1562

140 \*Australia \*dólar 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

1. Ingresar solicitantes: **[10]**

Si el usuario escoge esta opción, el sistema debe pedirle el nombre del archivo texto en el que se encuentran los datos básicos de los usuarios. Debe agregar al sistema cada usuario informando si pudo agregar o no el usuario. No debe permitir ingresar un usuario con el mismo número de pasaporte de otro usuario ya ingresado.

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 (por ejemplo “jardín”, “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 \*2017-12-30 \*gth@yahoo.es \*Yves Davoine

222 \*Yves Davoine \*Francia \*Grenoble \*1995-12-30 \*davoine@gmail.com

333 \*Gabriel Salas \*Chile \*Vina \*2012-05-21 \*gsal@hotmail.com \*Bachillerato

444 \*Keiko Tamura \*Japón \*Tokio \*1975-08-13 \*keiko@gmail.com

555 \*John Kidman \*Austria \*Viena \*2011-08-13 \*kidm@yahoo.com \*Primaria

#FIN

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

1. Ingresar tipos de visa y requisitos **[10]**

Si el usuario escoge esta opción, el sistema debe pedirle el nombre del archivo texto en el que se encuentran los requisitos de diferentes tipos de visas. Un ejemplo del archivo que contiene la información descrita es el siguiente:

#VISA

#tipoVisa

Turismo

#REQUISITOS

Dos fotos 4x5

Copia del pasaporte

Reserva de avión o de medio de transporte

Reserva hotelera

Certificaciones bancarias

Formulario de solicitud diligenciado

Seguro médico

Certificado laboral

#VISA

#tipoVisa

Estudiante

#REQUISITOS

Tres fotos 4x5

Copia del pasaporte

Reserva de avión o de medio de transporte

Domicilio en el país del sagrado corazón

Certificaciones económicas

Formulario de solicitud diligenciado

Seguro médico

Certificado de inscripción al programa

#VISA

#tipoVisa

Trabajo

#REQUISITOS

Dos fotos 4x5

Copia del pasaporte

Reserva de avión o de medio de transporte

Domicilio en el país del sagrado corazón

Certificaciones bancarias

Formulario de solicitud diligenciado

Seguro médico

Certificado laboral

Carta de acogida de la empresa

#VISA

#tipoVisa

Conyuge

#REQUISITOS

Dos fotos 4x5

Copia del pasaporte

Documento de identificación del cónyuge

Reserva de avión o de medio de transporte

Domicilio en el país del sagrado corazón

Certificaciones económicas personales

Certificaciones económicas del cónyuge

Formulario de solicitud diligenciado

Seguro médico

#FIN

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

1. Hacer solicitud de cita para visa de Turismo **[10]**

Si el usuario escoge esta opción, el sistema recibe por parte del usuario como parámetro, su número de pasaporte.

Para hacer una solicitud, el sistema sigue los siguientes pasos:

* Se debe crear un objeto de la clase Turismo. Para esto: se le pregunta al usuario por cuántos días va a estar (es decir, su estadía) y,
* Se busca en el archivo “tarifas.txt” el valor base de la visa, según el tipo. La estructura de dicho archivo 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

* Se asocia este objeto visa a la solicitud.
* Luego, se le pregunta al usuario si viaja sólo (ya tendría la información completa) o con alguien más. Si viaja sólo, el sistema debe buscar el usuario en el map de usuarios (dado su número de pasaporte) y seguiría el proceso de inscripción explicado anteriormente en la descripción dada antes del diagrama de clases. Vale la pena mencionar que se debe asociar dicho usuario a la solicitud, una vez ésta haya sido creada.
* Si viaja con alguien más, se solicita el nombre del archivo texto que contiene los números de pasaporte de sus acompañantes. Un ejemplo de dicho archivo es el siguiente:

#OTROS SOLICITANTES

#numPasaporte

111

333

555

888

#FIN

La convención #FIN indica que no hay más solicitantes (fin de datos) para esa solicitud.

* Crear la solicitud.
* Se asocia la visa a la solicitud
* El sistema debe buscar cada uno de los usuarios reportados en el archivo (en el map) y se agrega cada usuario a la solicitud que acaba de ser creada en el paso anterior.
* 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 signatura del método podría ser:

public void asignarFechaHora (Solicitud sol);

* Finalmente, el sistema asigna un número a dicha solicitud y se la muestra al usuario

1. Hacer solicitud de cita para visa diferente a Turismo **[10]**

Si el usuario escoge esta opción, el sistema recibe por parte del usuario como parámetro, su número de pasaporte.

Para hacer una solicitud, el sistema debe:

* Se pide el tipo de visa que desea solicitar
* Se crea una instancia de dicho tipo de visa. Se debe crear un objeto de la clase Trabajo o Estudio. Para esto:
  + Si es de Trabajo, se le pregunta al usuario la empresa y el cargo.
  + Si es de Estudiante, se le pide el nivel de escolaridad de los estudios que va a realizar y la institución donde los va a llevar a cabo
  + Se busca en el archivo “tarifas.txt” el valor base de la visa, según el tipo.
* Crear la solicitud.
* Se asocia este objeto visa a la solicitud.
* El sistema debe buscar al usuario y se agrega dicho usuario a la solicitud que acaba de ser creada en el paso anterior.
* Se asigna una fecha y hora para la cita.
* Finalmente, el sistema asigna un número a dicha solicitud y se la muestra al usuario

1. Calcular valor de visa: **[20]**

Si el usuario escoge esta opción, el usuario ingresa su número de pasaporte, o el número de solicitud, 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.

* Si el usuario es un Niño0-2, sólo paga el 10% de la tarifa de la visa.
* Si el usuario es un Niño2-12:
  + La visa tiene un 5% de descuento por cada año que le falte para los 18 años.
  + Si el tipo de Visa es de Turismo, tiene un incremento del 20%.
  + 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:
  + La visa tiene un 25% de descuento.
  + Si el tipo de Visa es de Turismo, tiene un incremento del 10%.

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 signatura de dicho método podría ser:

public static int edadEnAnnos

(LocalDate f1, LocalDate f2);

La interacción con el usuario debe ser como se ilustra a continuación (puede que los cálculos de la visa no correspondan a los valores reales, sólo se muestra la organización del archivo de reporte):

--Calcular Valor Visa para embajada de Austria

----Favor indicar el número antecedido por CP si consulta por código de solicitud o NP si consulta por número de pasaporte:

**NP222**

\*\* Numero de pasaporte valido

----Se muestran los datos del usuario en el sistema:

#numPass—--nombre-----paisOrigen------ciudadNac------fechaNac----email

222 Yves Davoine Francia Grenoble 1995-12-30 davoine@gmail.com

----Su visa es de:

Turismo

----Se muestran los solicitantes asociados con el valorVisa en pesos:

#numPass—--nombre-----fechaNac----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

333 Gabriel Salas 1960-05-21 $150.000 $37.500 $187.500

444 Keiko Tamura 1975-08-13 $150.000 $37.500 $187.500

555 John Kidman 2008-08-13 $120.000 $30.000 $150.000

----El valor total de la visa es:

PSC—------Tasa de cambio ------Euros

862.500 $2550 339

1. Reporte de citas para una determinada fecha **[10]**

Si el usuario escoge esta opción, el usuario escribe una fecha (aaaa-mm-dd) y el sistema debe generar un reporte con los usuarios, número de pasaporte y de solicitud. Para cada usuario se debe mostrar sus datos propios (número de pasaporte, nombre completo, tipo de visa a solicitar) y el número de solicitud. El reporte debe estar ordenado por nombre de solicitante, de manera ascendente. Un ejemplo del reporte es el siguiente:

--REPORTE DE SOLICITUDES EMBAJADA DE AUSTRIA

Fecha: 15-03-2014

#numPass—-----nombre-----------tipoVisa----numSolicitud

286974 Aurelie Arnaud Trabajo 256897

333 Gabriel Salas Turismo 552211

111 Grettel Lahmann Turismo 552211

212055 Isabela Mendoza Estudiante 452368

555 John Kidman Turismo 552211

444 Keiko Tamura Turismo 552211

222 Yves Davoine Turismo 552211

Además, se debe generar un archivo texto cuyo nombre es: **Citas-fecha.txt**.

Para el ejemplo anterior, el archivo se debe llamar: **“Citas-15-03-2014.txt”**

1. Consultar la lista de requisitos de un tipo de Visa **[5]**

Si el usuario escoge esta opción, el usuario selecciona el tipo de visa del que desea conocer los requisitos**. La lista de tipos de visa se debe presentar ordenada alfabéticamente.**

Luego, el sistema le muestra los requisitos, **también ordenada alfabéticamente según su descripción**.

La interacción con el usuario debe ser como se ilustra a continuación:

--Lista de requisitos para Visa

--Tipos de Visa

Conyuge

Estudiante

Trabajo

Turismo

----Favor indicar el tipo de visa del que desea conocer los requisitos:

**Turismo**

\*\* Tipo de visa valido

----Requisitos:

Certificaciones bancarias

Certificado laboral

Copia del pasaporte

Dos fotos 4x5

Formulario de solicitud diligenciado

Reserva de avión o de medio de transporte

Reserva hotelera

Seguro médico

1. Consultar la lista de beneficiados **[10]**

Si el usuario escoge esta opción, 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. Además, se debe imprimir el total de lo que dejó de recaudar la embajada por concepto de los valores de la visa.

**La lista de beneficiarios debe estar ordenada alfabéticamente por nombre**. Dicho reporte se debe tanto mostrar por pantalla como generar un archivo llamado “**beneficiarios.txt**”

La interacción con el usuario debe ser como se ilustra a continuación (puede que los cálculos de la visa no correspondan a los valores reales, sólo se muestra la organización del archivo de reporte):

--Lista de beneficiarios

#numPass—--nombre----- valorTotal (pesos)

222 Yves Davoine $187.500

111 Grettel Lahmann $150.000

333 Gabriel Salas $187.500

444 Keiko Tamura $187.500

555 John Kidman $150.000

El valor total en pesos que se dejó de recaudar por visas de beneficiados es: $862.500

**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 **domingo 15 de abril de 2018 hasta las 11:59 pm de la noche.**

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

1. **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.

1. **Entregables**

Un solo archivo .zip nombrado de la siguiente manera: **proy2-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 TestCitasEmbajada (archivos .class)
* Archivo ejecutarTest.bat que ejecuta el jar anterior
* Archivo doc.zip con la documentación javadoc del sistema

1. No es necesario entregar los archivos de datos. La revisión se hará con archivos consistentes colocados en la raíz del proyecto.
2. **Observaciones**

* Si no se entregan los archivos enumerados en el punto 3, 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
* 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 y bajo cualquier ambiente de ejecución (incluyendo la ejecución por consola).
* Se deben cumplir las reglas de Juego del curso.