



Universidad Católica del  
Norte Sede Coquimbo  
Escuela de Ingeniería

## TALLER N°2 - PROGRAMACIÓN AVANZADA

**Docentes:** Loreto Telgie Bendek – Cristian Chiang Ramírez – Carlos Luco Montofré.

**Ayudante:** José Espinoza

---

Fecha Entrega del Enunciado : Jueves 14 de septiembre.

Fecha de Término : Lunes 16 de octubre. Hasta las 23:59 hrs.

---

### DOMINIO DEL PROBLEMA

La biblioteca de la universidad “ABC” está solicitando ayuda de los alumnos de programación avanzada y programación I, para desarrollar un sistema que facilite la gestión de la unidad, ya que el actual sistema informático no entrega esas facilidades. Con este objetivo, los administradores de la biblioteca exportaron a un archivo de texto todos los datos que han sido almacenados hasta el momento. Los archivos son **clientes.txt**, **libros.txt**, **prestamos.txt**.

A partir de los datos que se han exportado y de los requerimientos que han sido identificados, se conoce que existen clientes (rut, nombre, apellido, número de celular y el correo electrónico de contacto). Además, se han exportado todos los libros que están presente en la biblioteca. De cada libro se conoce, el identificador ISBN, el título, el nombre del autor y el año de publicación, además de estar asociado a una lista de préstamos.

Cada cliente puede pedir un préstamo de algún libro que esté presente en biblioteca, por lo que tendrá una lista de préstamos asociada. Estos préstamos se registraron en el archivo **prestamos.txt**; donde cada préstamo posee un código único, la identificación del cliente que ha pedido el préstamo, el libro pedido, la fecha del préstamo, la fecha de devolución estimada y la fecha de devolución real. Una de los aspectos que necesita conocer de la biblioteca es la deuda de cada cliente (que pudo haber sido pagada o no), la cual se calcula de la siguiente forma:

La deuda se calcula como el **N° de días \* factor de la deuda**, donde el factor de la deuda se calcula como la diferencia en días entre la fecha de devolución real y la fecha de devolución estimada, y si:

1. Si lo ha entregado en 1 a 7 días, entonces el factor de la deuda es \$100.
2. Si lo ha entregado en 8 días a 14 días, entonces el factor de la deuda es \$300.
3. Si lo ha entregado en 15 días o más, entonces el factor de la deuda es de \$700.

No todos los clientes que han pedido préstamos han devuelto los libros, por lo que ese préstamo no registrará ninguna deuda asociada. Para saber si un cliente ha devuelto el libro, se registrará en el archivo **prestamos.txt** la siguiente información:

#### Registro de libro pedido y devuelto

1, 17679133-0, 123-4, 31-08-2017, 05-09-2017, 08-09-2017 ← Devuelto 3 días después (deuda de \$300)

#### Registro de libro pedido y no devuelto

1, 17679133-0, 123-4, 31-08-2017, 05-09-2017, 0 ← Aún no ha devuelto el libro. (todavía no registra deuda)

## REQUISITOS FUNCIONALES

La biblioteca requiere que el sistema desarrollado en Java cumpla con los siguientes requisitos funcionales:

**RF1:** Dado el rut de un cliente ingresado por consola, imprimir a) Para cada préstamo que genera una deuda a dicho cliente (si el libro lo devuelve cuando corresponde o aún no lo devuelve, entonces no genera deuda), el código del préstamo, la fecha del préstamo, la fecha de devolución estimada, la fecha de devolución real, la diferencia en días entre la fecha de devolución real y la estimada, y la deuda asociada. b) La deuda total registrada.

**RF2:** Dado un rango de fechas que se han ingresado por consola (Por ej. "26-09-2017" y "29-09-2017"), imprimir el libro que más ha sido solicitado durante ese rango; el ISBN del libro, el título, el nombre de su autor, el año de publicación y la cantidad de veces que ha sido solicitado.

**RF3:** Generar el archivo **devoluciones\_pendientes.txt** que contenga: el código único del préstamo, la fecha del préstamo, la fecha de devolución estimada, el apellido del cliente, el nombre del cliente, el n° de celular y correo electrónico del cliente, y el título del libro, para todos los préstamos en que aún no se ha devuelto el libro. Debe estar ordenado por fecha de devolución estimada de la más antigua a la más reciente.

**RF4:** Generar un menú con opciones para cumplir con RF1 - RF3 y una opción para finalizar la ejecución del programa.

## CONSIDERACIONES

- Se les proporcionarán los archivos de texto con los que trabajará
- Un año tiene 365 días (No considere años bisiestos)
- Ambos archivos de texto entregados tendrán sus datos separados por una coma (,).
- **Los archivos de texto solo se deben leer una vez durante el proceso.**
- El taller debe ser realizado en grupos de 2 alumnos del mismo paralelo.
- Para la entrega 1 y 2 se debe publicar en la plataforma el informe en formato .pdf y entregar impreso.
- Para la entrega 3 se debe publicar en la plataforma el proyecto (exportar como .zip), el cual llevará por nombre el o los apellidos del o los integrantes. Por ejemplo: castro.rojas.zip
- El nombre del Proyecto se denominará Taller2.
- Se debe hacer uso de subprogramas (funciones y/o procedimientos) y POO.
- La entrega 3 debe realizarse en NetBeans o Eclipse.
- El taller deberá estar documentado. Se recomienda utilizar el estándar Java Doc.
- Las consultas se realizarán al correo del ayudante. (**Ayudante:** José Espinoza, **correo:** jes004@alumnos.ucn.cl)
- Cada incumplimiento será sancionado con 0,3 de la calificación final.
- En caso de copia (códigos completamente iguales) la nota será 1.0 para todos los que entregaron el mismo código y el caso se presentará ante el consejo.

## FECHAS DE ENTREGA

- **Entrega N° 1: 27 de septiembre (23:59)**, Modelo de Dominio y Contratos.  
Se debe entregar un informe (pdf e impreso) con el modelo de dominio y los contratos.
- **Entrega N° 2: 04 de octubre (23:59)**. Diagrama de Clases.  
Se debe entregar un informe (pdf e impreso) del Diagrama de Clases.
- **Entrega N° 3: 16 de octubre (23:59)**. Código de la aplicación.  
Se debe subir a plataforma un archivo .zip que contemple la **carpeta del proyecto** donde está almacenado todo el código fuente de la aplicación realizado en NetBeans o Eclipse.

La revisión va a considerar

- Si el archivo subido no es un .zip o un .rar (o similar), la nota máxima es un 2.0
- El .zip o .rar debe contener la **carpeta del proyecto** (comunicarse si tiene dudas), en el caso que únicamente el archivo .java, entonces la nota máxima será un 4.0
- Si el código no compila, la nota máxima es un 2.0