



# Proyecto final: Salvando la biblioteca de la sede Fundamentos de Programación

Carlos Andres Delgado S, Msc  
`carlos.andres.delgado@correounivalle.edu.co`

Febrero de 2020

## Reglas del proyecto

1. Este proyecto trabajado en grupos de hasta 3 personas.
2. Este proyecto debe ser entregado el **Sábado, 7 de Marzo a las 23:55** hora de Colombia, por el campus virtual.
3. Se solicita respete los nombres de funciones especificados en el proyecto.
4. Debe entregar el código fuente organizado en carpetas dentro del primer nivel del archivo comprimido, no cree una jerarquía compleja difícil de revisar. El no cumplimiento de esta regla traerá sanciones a su nota.
5. No se permite copiar código de Internet ni de sus compañeros. Si se encuentra código copiado el taller será anulado por completo.
6. Entregue un sólo archivo comprimido. No entregue archivos comprimidos dentro de archivos comprimidos, ya que esto dificulta la revisión enormemente. El no cumplimiento de esta regla traerá sanciones a su nota.
7. No insertar comentarios en caja ni pegar imágenes, para que el archivo se guarde en formato de texto. Esto se debe a que el docente usa scripts para pruebas automáticas, no cumplir este punto traerá sanciones a su nota.
8. Las primeras líneas de cada archivo de código fuente, deben tener los integrantes del grupo con sus nombres y código completos. El no cumplimiento de esta regla traerá sanciones a su nota.
9. Debe dejar en el código 3 ejemplos de ejecución de cada una de las funciones principales que diseñe (no es necesario de las auxiliares), no hacerlo será sancionado.

# 1. Problema: Virtualizando la biblioteca de Tulua

Es el año de 1996 y recién están entrando las computadoras a Colombia, usted se ha vuelto amigo de la persona encargada de biblioteca y esta le comenta que están teniendo dificultades en hacer seguimiento de los prestamos de libros, debido a que se llevan en cuadernos y estos se han mojado por una guerra de pistolas de agua que se presentó la semana pasada.



Figura 1: Biblioteca de la sede, tomada el archivo histórico de la sede Tulua - Universidad del Valle

Usted que es ingeniero de sistemas recién graduado, le explica a la encargada que puede desarrollar un sistema para el préstamo de libros, mediante un lenguaje de programación conocido como Racket.

**Nota honorífica:** Este proyecto es en honor a las personas que terminaron mojadas en la guerra de pistolas de agua del 96, así mismo a los 3 gatos que resultaron afectados. Y también de las personas que pararon para avanzar en el año 2019.

## 1.1. Elementos de la biblioteca virtual versión 96

En una entrevista realizada a la encargada de la sede, esta le comenta que los elementos de la biblioteca son:

Un libro puede tener:

- Un titulo, que es una cadena de texto
- Uno o más autores, que son cadenas de texto
- Lugar de publicación, que es una cadena de texto que contiene el nombre del país.
- Año de publicación, que es un número

- ISBN, que es un código numérico
- Una o más materias que cubren
- Clasificación: La cual tiene dos campos: un número que indica el estante y otro numero que indica su posición en el estante.
- Número de ejemplares disponibles de la sede

La información de un libro es por ejemplo:

Título How to design programs: an introduction to programming and computing  
 Autor(es) Felleisen, Matthias (Autor)  
 Findler, Robert Bruce (Autor)  
 Flatt, Matthew (Autor)  
 Krishnamurthi, Srikanth, 1969- (Autor)  
 Publicación Estados Unidos : Massachusetts Institute of Technology, 2001  
 ISBN 0262062186  
 Clasificación(es) 005.12  
 Materia(s) Programación de computadores (Computadores electrónicos);  
 Procesamiento electrónico de datos; Páginas Web;  
 Número de ejemplares: 7

Puede obtener información aquí: <http://opac.univalle.edu.co/cgi-olib/>

Usted decide cómo codifica esta información en los archivos de texto, de acuerdo a la librería de carga de archivos de Racket.

Un usuario de la biblioteca, puede ser estudiante, profesor o funcionario, los cuales:

- Si es estudiante, tiene un nombre, un programa de estudios y un código
- Es profesor, tiene un nombre y una cédula
- Si es funcionario, tiene un nombre, una cédula y una dependencia de trabajo

Estos pueden solicitar prestados uno o más libros, los cuales deben ser asignados a ellos.

El requerimiento de la aplicación son los siguientes:

1. Cargar usuarios y libros disponibles en la biblioteca desde un archivo de texto. Los usuarios inicialmente no tienen libros prestados.
2. Buscar libros por su titulo, por ISBN, por autor o por materia y almacenar la búsqueda en un archivo de texto
3. Buscar usuarios por cédula o código.
4. Prestar un libro a un usuario y modificando la disponibilidad de ejemplares de la biblioteca.
5. Almacenar en un archivo la información actualizada de usuarios con sus prestamos y de los libros.

## 1.2. Evaluación

Para la justificación de los puntos diseñe un informe sencillo en formato PDF, el cual debe contener el nombre de los integrantes del grupo.

1. [15 puntos] Definición de estructuras de datos para la gestión de la biblioteca, justifique cada una de ellas.
2. [10 puntos] Diseñe la función **cargar-datos** esta función recibe dos cadenas de texto que indican las rutas de los archivos que contienen los libros y usuarios, ejemplo:

```
(cargar-datos "data/libros.txt" "data/usuarios.txt")
```

Explique la estrategia de carga y almacenamiento de esta información. Genere archivos que al menos tengan 6 usuarios y 15 libros respectivamente.

3. [20 puntos] Diseñe la función **buscar-libro** la cual permite buscar por autor, materia ó ISBN, bajo la siguiente plantilla:

```
(define (buscar-libro f)
  ...
)

(define (buscar-autor autor)
  (buscar-libro (lambda (x nom) ...))
)

(define (buscar-ISBN codisbn)
  (buscar-libro (lambda (x codisbn) ...))
)

(define (buscar-materia mat)
  (buscar-libro (lambda (x mat) ...))
)
```

Este retorna la información de los libros encontrados, usted define como se muestra esa información.

4. [15 puntos] Diseñe la función **buscar-usuario** el cual recibe una cédula o código, esta debe seguir el siguiente formato:

```
(define (buscar-usuario num)
  ...
)
```

Usted define cómo se retorna la información del usuario encontrado. Así mismo debe mostrarse la información de los libros prestados al usuario

5. [30 puntos] Diseñe la función **prestar-libro** la cual recibe un cédula o código del usuario y el autor, materia ó ISBN del libro a prestar. Debe seguir la siguiente plantilla

```
(define (prestar-libro usuario libro)
  ...
)
```

Debe tomar en cuenta lo siguiente:

- Debe agregarse al usuario el libro prestado, un usuario puede tener cero o más libros prestados.
- Debe actualizar el campo número de ejemplares de los libros
- Considere los casos donde el usuario o libro no existen
- Los cambios son permanentes, es decir que si se almacena la información de usuarios o libros debe considerar este cambio.

Para esto explore las funciones `begin` y `set!` de Dr Racket.

**Nota:** Ya que la búsqueda de libros puede dar uno o más libros, tome el primero de los resultados como el libro a prestar.

6. [10 puntos] Diseñe las funciones **almacenar-libros** y **almacenar-usuarios**, estas toman los datos **actualizados** de los usuarios y libros para almacenarlos en un archivo. siga los siguientes ejemplos:

```
(almacenar-libros "data/libros-update.txt")
;;Crea el archivo update.txt con la información de los libros

(almacenar-usuarios "data/usuarios-update.txt")
;;Almacena la información de los usuarios tomando en cuenta que
  libros han prestado.
```

**Nota:** Los ejemplos suponen que existe una carpeta `data` dentro de su proyecto, sin embargo puede guardarlos en el lugar que considere más apropiado.

Tenga en cuenta que el proyecto debe ser sustentado, la nota de sustentación es individual y va entre 0 y 1, la cual es multiplicada por la nota del proyecto. Por ejemplo, si su grupo obtuvo 5.0 y su nota es 0.5, la nota total del proyecto para usted será 2.5.