CS2042

BASES DE DATOS II

File Organization

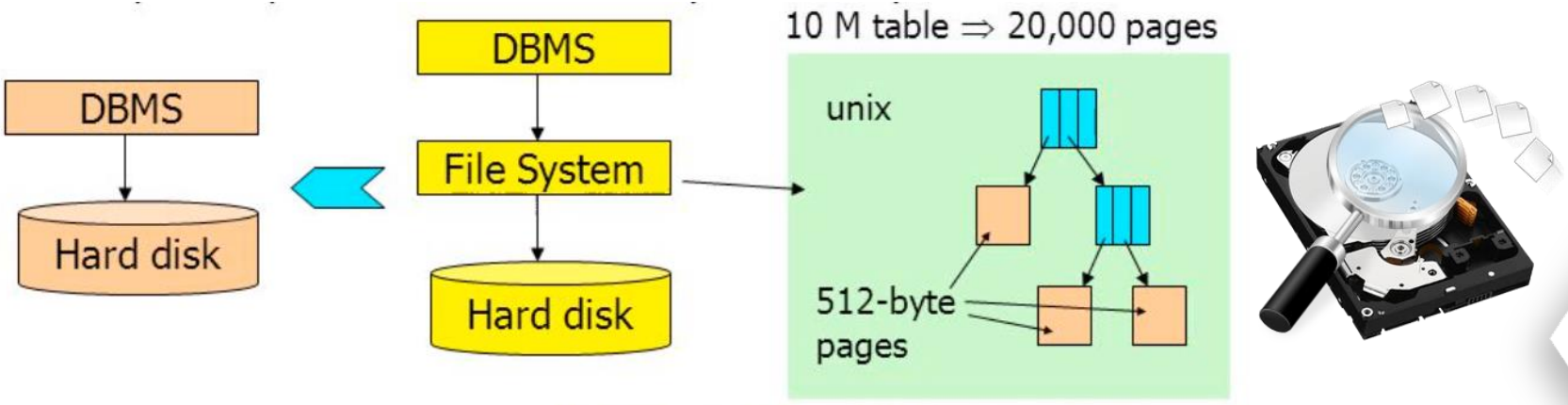
Introducción

Semana 01

Heider Sanchez
hsanchez@utec.edu.pe

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Organización de Archivos

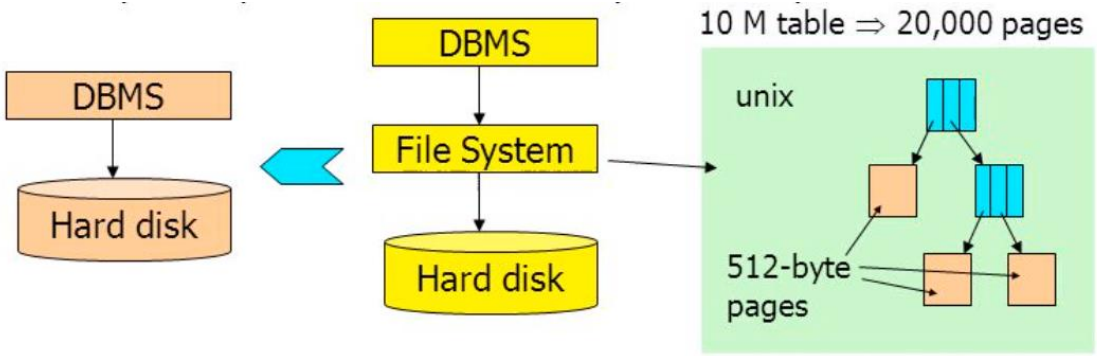


Indexes

Organización de Archivos

- Una base de datos es almacenado como una **colección de archivos**.
- Cada archivo es una secuencia de **registros**.
- Un registro es una secuencia de **campos**.

  - un archivo ⇔ una tabla
  - un registro ⇔ una tupla
  - un campo ⇔ una columna



Conceptos

- **Field:** Esta es la unidad más pequeña de almacenamiento, también conocido como atributo o columna. Un campo tiene dos propiedades: nombre y tipo de dato. El tipo define el tamaño en bytes que requiere su almacenamiento.

- **Record :** Es una colección de campos, también conocido como tupla o fila. Por ejemplo, un registro de empleado puede consistir en los siguientes campos: IdEmpleado, Nombre, Dirección, etc.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Conceptos

- **File:** Es un conjunto relacionado de registros, también conocido como tabla o relación. Un archivo tiene ciertas propiedades como su nombre, tamaño y localización. Los archivos pueden ser archivos de texto o binarios:
  - Los archivos de texto almacenan números como una secuencia de caracteres
  - Los archivos binarios almacenan números en formato binario.

Conceptos

- **File Organization:** Un archivo tiene dos facetas: logico y fisico. El archivo lógico es todo el conjunto de registros en modo de [tabla]. Mientras que el físico muestra como los registros son almacenados físicamente en disco.

  Organización de Archivos se refiere a la representación física de un archivo en disco.

Conceptos

- **Key:** Este es un atributo que identifica de manera única a cada registro. Similar a la clave primaria de una tabla en una base de datos.
- **Page:** Un archivo es transferido hacia la memoria principal para realizar operación como inserción, modificación eliminación, búsqueda, etc. Si el archivo es demasiado grande, este es fragmentado en páginas de igual tamaño (bloques), el cual es la unidad de intercambio entre el disco y la memoria principal.
- **Index:** Este un puntero a un registro en un archivo, el cual provee acceso eficiente y rápido a los registros.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Operaciones básicas en archivos

- **Read:** Es es el proceso de leer registros desde un archivo, de esta manera se pueden efectuar las operaciones necesarias sobre los campos del registro.
- **Write:** Es el proceso de escribir nuevos registros a un archivo. Pueden ser agregados al final del archivo o insertados en una posición determinada.
- **Delete:** Remueve registros del archivo.

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Operaciones básicas en archivos

- **Read:** Es es el proceso de leer registros desde un archivo, de esta manera se pueden efectuar las operaciones necesarias sobre los campos del registro.
- **Write:** Es el proceso de escribir nuevos registros a un archivo. Pueden ser agregados al final del archivo o insertados en una posición determinada.
- ~~**Delete:** Remueve registros del archivo.~~

Operaciones básicas en archivos en C++

Class	Contents
filebuf	Its purpose is to set the file buffers to read and write. Contains **Openprot** constant used in the **open()** of file stream classes.Also contain **close()** and **open()** as members.
fstreambase	Provides operations common to file streams.Serves as a base for **fstream,ifstream** and **ofstream** class.Contains **open()** and **close()** functions.
ifstream	Provides input operations.Contains **open()** with default input mode.Inherits the functions **get(),getline(),read(),seekg(),tellg()** functions from istream.
ofstream	Provides output operations.Contains **open()** with default output mode.Inherits **put(),seekp(),tellp()** and **write()** functions from **ostream**.
fstream	Provides support for simultaneous input and output operations.Contains open with default input mode.Inherits all the functions from **istream** and **ostream** classes through **iostream**.

https://www.geeksforgeeks.org/file-handling-c-classes/
http://www.cplusplus.com/doc/tutorial/files/

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

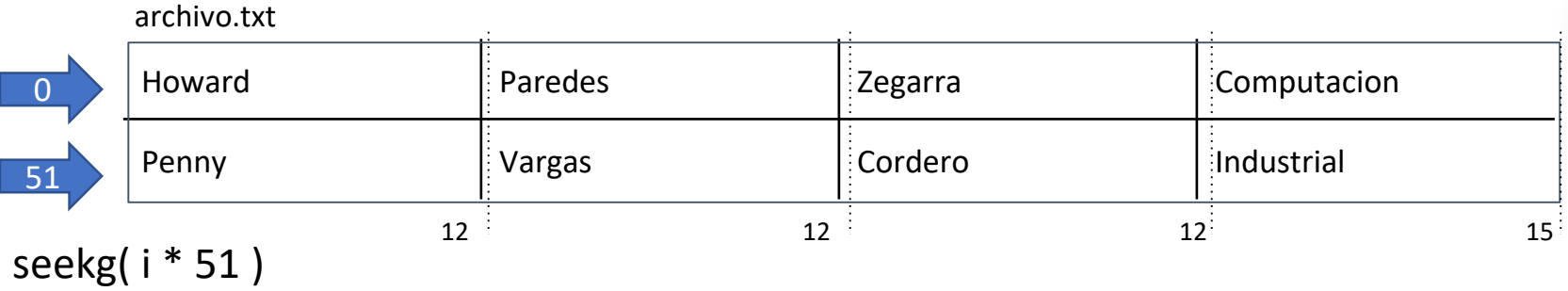
Organizando Registros en un Archivo

1. Fixed-Length Records
2. Variable-Length Records

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Fixed-Length Records

- Todos los registros en un archivo tienen la misma longitud y la misma cantidad de campos. El tamaño de cada campo es el mismo para cada registro. Esto asegura fácil ubicación de los valores de cada campo ya que sus posiciones están predeterminadas.



Fixed-Length Records: Escritura

Record

```
class Alumno
{
public:
  char Nombre [12];
  char Apellidos [12];
};
```

Escribir un Registro

```
ostream & operator
  << (ostream & stream, Alumno & record)
{
  stream.write(record.Nombre, 12);
  stream.write(record.Apellidos, 12);
  stream << "\n";
  stream << flush;
  return stream;
}
//Archivo binario
outFile.open(“archivo.dat”,ios::app | ios::binary)
outFile.write((char*) &record, sizeof(record))
```

Fixed-Length Records: Lectura

Record

```
class Alumno
{
public:
  char Nombre [12];
  char Apellidos [12];
};
```

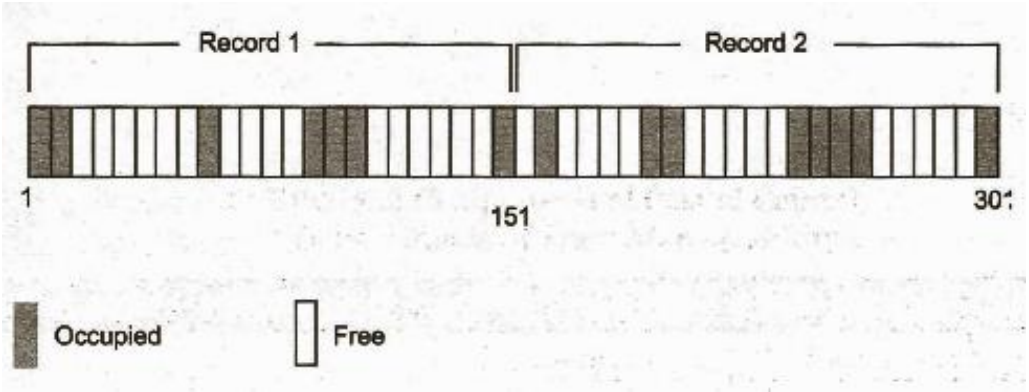
Leer un Registro

```
istream & operator
  >> (istream & stream, Alumno & record)
{
  stream.read(record.Nombre, 12);
  stream.read(record.Apellidos, 12);
  stream.get(); //read \n
  return stream;
}
//Archivo binario
inFile.open("archivo.dat",ios::in | ios::binary)
inFile.read((char*) &record, sizeof(record))
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Fixed-Length Records: Acceso Directo

- Desde que cada registro ocupa espacios iguales, identificar el inicio y el fin de cada registro es relativamente simple.




```
Alumno record;
n = sizeof(record);
start = n * i;
end = start + n;
```

i: posición lógica del registro
start: posición física del registro

Fixed-Length Records: Acceso Directo

Record

```
class Alumno
{
public:
  char Nombre[12];
  char Apellidos [12];
};
```

Leer un Registro

```
Alumno readRecordBin(int i)
{
 ifstream inFile;
 Alumno record;
 inFile.open("file.dat", ios::binary);
 inFile.seekg(i * sizeof(record), ios::beg);
 inFile.read((char *) &record, sizeof(record));
 inFile.close();
 return record;
}
```

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Fixed-Length Records

- **Problemas**:
  - El acceso a los registros es simple pero se corre el riesgo de cruzar bloques.
  - La modificación no permite que los registros crucen el límite del bloque.
  - ¿Eliminación de registros?

	Id	Nombre	Ciclo
1	P-102	Andrea	5
2	P-251	Carlos	7
3	P-412	Maria	3
4	P-255	Juan	7
5	P-250	Javier	7
6	P-312	Mabel	3
7	P-982	Saulo	9

Fixed-Length Records: Eliminación

- **Alternativa 1**:
  - Mover los registros *i+1,...,n* hacia *i,...,n-1*

¿En donde mantener el size?

¿Complejidad?

	Id	Nombre	Ciclo
1	P-102	Andrea	5
2	P-251	Carlos	7
3	P-412	Maria	3
4	P-255	Juan	7
5	P-250	Javier	7
6	P-312	Mabel	3
7	P-982	Saulo	9

Fixed-Length Records: Eliminación

- **Alternativa 1**:
  - Mover los registros *i+1,...,n* hacia *i,...,n-1*

¿En donde mantener el size?

¿Complejidad?

	Id	Nombre	Ciclo
1	P-102	Andrea	5
2	P-412	Maria	3
3	P-255	Juan	7
4	P-250	Javier	7
5	P-312	Mabel	3
6	P-982	Saulo	9
7	P-982	Saulo	9

Fixed-Length Records: Eliminación

- **Alternativa 2**:
  - Mover el registro *n* hacia *i*

¿Complejidad?

	Id	Nombre	Ciclo
1	P-102	Andrea	5
2	P-251	Carlos	7
3	P-412	Maria	3
4	P-255	Juan	7
5	P-250	Javier	7
6	P-312	Mabel	3
7	P-982	Saulo	9

Fixed-Length Records: Eliminación

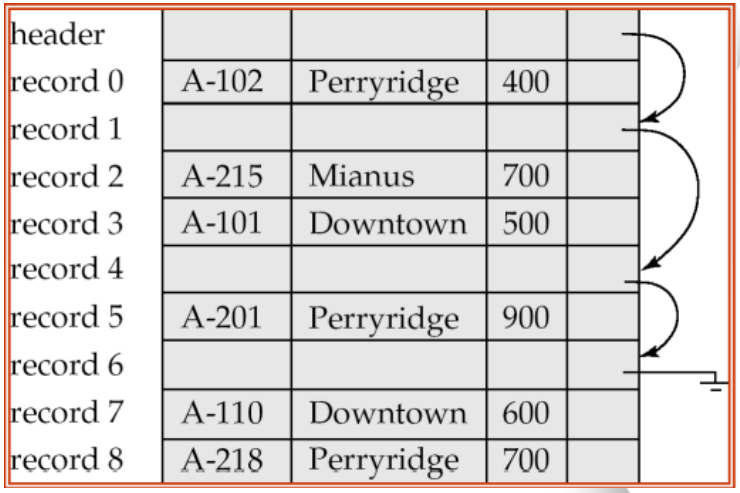
- **Alternativa 2**:
  - Mover el registro *n* hacia *i*

Size = 5

	Id	Nombre	Ciclo
1	P-102	Andrea	5
2	P-982	Saulo	9
3	P-312	Mabel	3
4	P-255	Juan	7
5	P-250	Javier	7
6	P-312	Mabel	3
7	P-982	Saulo	9

Fixed-Length Records: Eliminacion

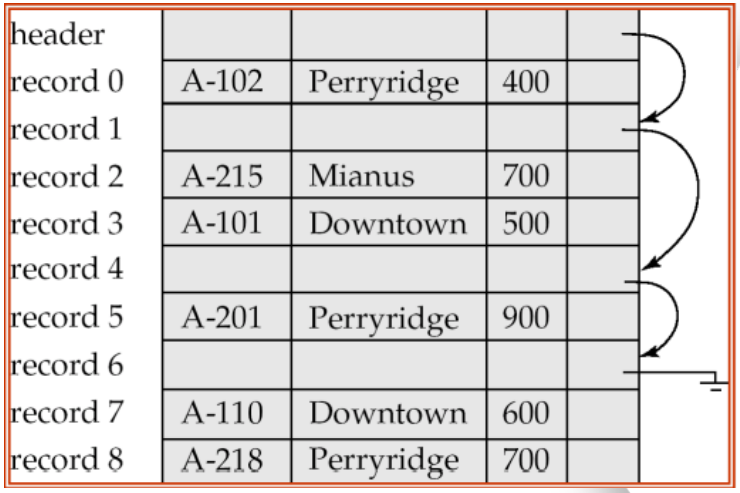
- **Alternativa 3**:
  - No mover registros, pero enlazar todos los registros liberados en una lista (Free List).



Free List: eliminar registros 6,4 y 1.

Fixed-Length Records

- **Free List**:
  - Almacenar la dirección del primer registro eliminado en el **header**
  - Utilice este primer registro para almacenar la dirección del segundo registro eliminado, y así sucesivamente.
  - Podemos asumir estas direcciones almacenadas como punteros, ya que "apuntan" a la ubicación de un registro.
  - **Optimización**: se puede usar los mismo registros eliminados para guardar los punteros.



Free List: eliminar registros 6,4 y 1.

- **Free List:**

datos.dat

-1

	Id	Nombre	Ciclo	NextDel
1	P-256	Federico	1	0
2	P-251	Carlos	7	0
3	P-412	Maria	3	0
4	P-255	Juan	7	0
5	P-250	Javier	7	0
6	P-312	Mabel	3	0
7	P-982	Saulo	9	0

Eliminar 3, 5 y 1

	Eliminación	Inserción
FIFO	O( )	O( )
LIFO	O( )	O( )

* k es total de eliminados

Variable-Length Records

archivo.csv

```
Heider,Sanchez,CS
Juan,Vaca del Campo,CS
Maria,Lopez,CS
Ana,Sanchez,CS
Karla,Sanchez,CS
```

Se necesita guardar:

1- separador de campo

2- separador de registro

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Variable-Length Records

- El manejo de archivos con registros de longitud variable surgen en los sistemas de bases de datos para dar soporte a los campos con longitud variable (TEXT, JSON, BLOB, etc).
- Ventaja: se hace uso de la memoria (RAM y Secundaria) de manera más eficiente.
- Para determinar el inicio y el final de cada campo o registro, se requiere de separadores especiales:
  - Usando caracteres como delimitadores.
  - Usando indicadores de longitud de campo/registro.

Variable-Length Records

1. Archivos de Texto: Usando caracteres especiales para separar los campos.

- Dichos caracteres no deben aparecer en ningún lugar dentro del valor del campo.
- La ubicación de cualquier campo dentro del registro requiere una exploración del registro hasta que se encuentre el campo.

```
Howard|Paredes|Zegarra|Computacion|5|1500.50
Penny|Vargas|Cordero|Industrial|2|2850.00
```

← \n
como separador de registro

Problemas:

- El delimitador es parte del contenido.
- Acceso directo a un registro. O(n)
- Eliminar un registro. O(n)

Variable-Length Records

2. Archivos Binarios: Usando indicadores de longitud de campo/registro.

- Dicho indicador debe estar al inicio del campo o del registro para ser leído primero.
- Sólo se requiere indicar el tamaño de los campos de tipo texto.

```
43:6:Howard7:Paredes7:Zegarra11:Computacion4:58:1500.50
40:5:Penny6:Vargas7:Cordero10:Industrial4:28:2850.00
```

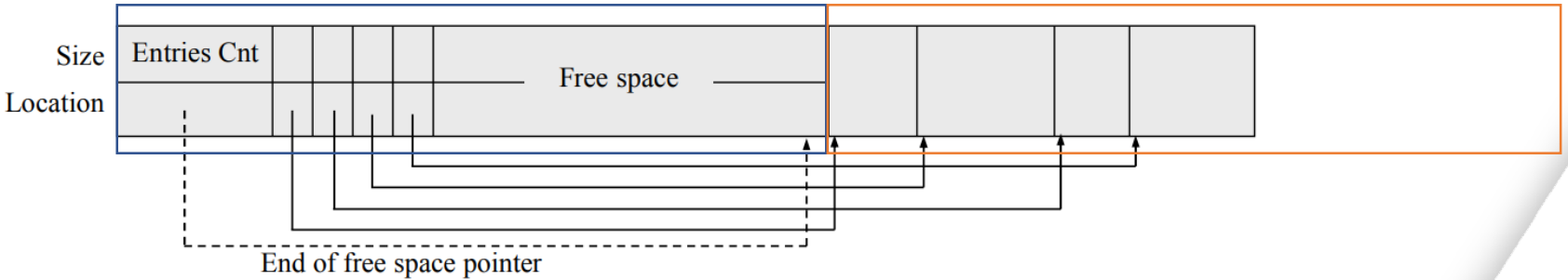
Problemas:

- ~~El delimitador es parte del contenido~~
- Acceso directo a un registro
- Eliminar un registro

https://www.cs.scranton.edu/~mccloske/courses/cmps340/file_record_storage.html

Variable-Length Records

3. Slotted Page: cabecera que indica el inicio de cada registro



- Slotted Page contiene:
  - Localización y tamaño de cada registro.
  - *El número de registros de entrada.*
    - *El final del espacio libre separado para este encabezado.*
- Para localizar un registro siempre se verifica el encabezado
- Mantener actualizado el encabezado

http://labe.felk.cvut.cz/~stepan/AE3B33OSD/Lesson10-Data_Access.pdf

Variable-Length Records

Cabecera.dat

	Posición	Tamaño
1	0	18
2	18	21
3	39	20
4	59	24
5	83	24

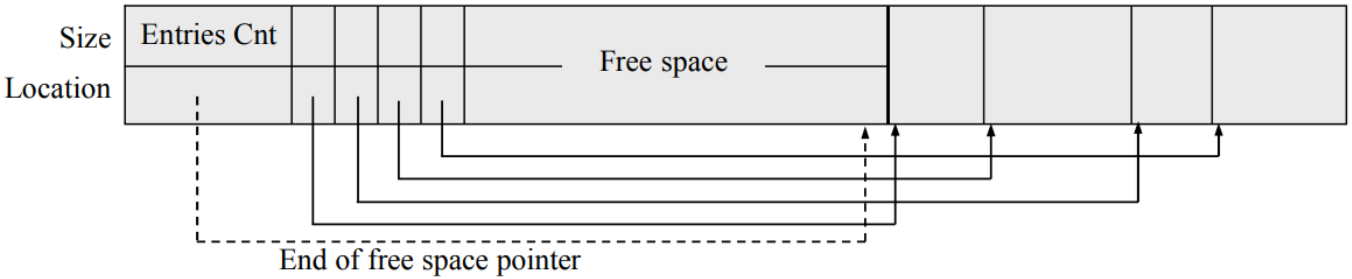
Datos.txt

		Codigo|Nombre|Apellidos|Carrera
seekg(0)	1	001|Jose|Lopez|CS
seekg(18)	2	002|Maria|Vergara|IN
seekg(39)	3	003|Luis|Vergara|IN
seekg(59)	4	004|Patricia|Vergara|IN
seekg(83)	5	005|Valentin|Vergara|IN

Leer el registro i → T(n) = 1 + 1 = 2 → O(1)

Variable-Length Records

3. Slotted Page: cabecera que indica el inicio de cada registro



Problemas:

- ~~El delimitador es un carácter~~
- ~~Acceso directo a un registro~~
- **¿Eliminar un registro?**

Variable-Length Records

ENUNCIADO DE LABORATORIO

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA