

Universidad Rafael Landívar

Facultad de Ingeniería

Laboratorio de Estructuras de Datos II

Sección 02

Ingeniero Mario Rodrigo Hernández Soto



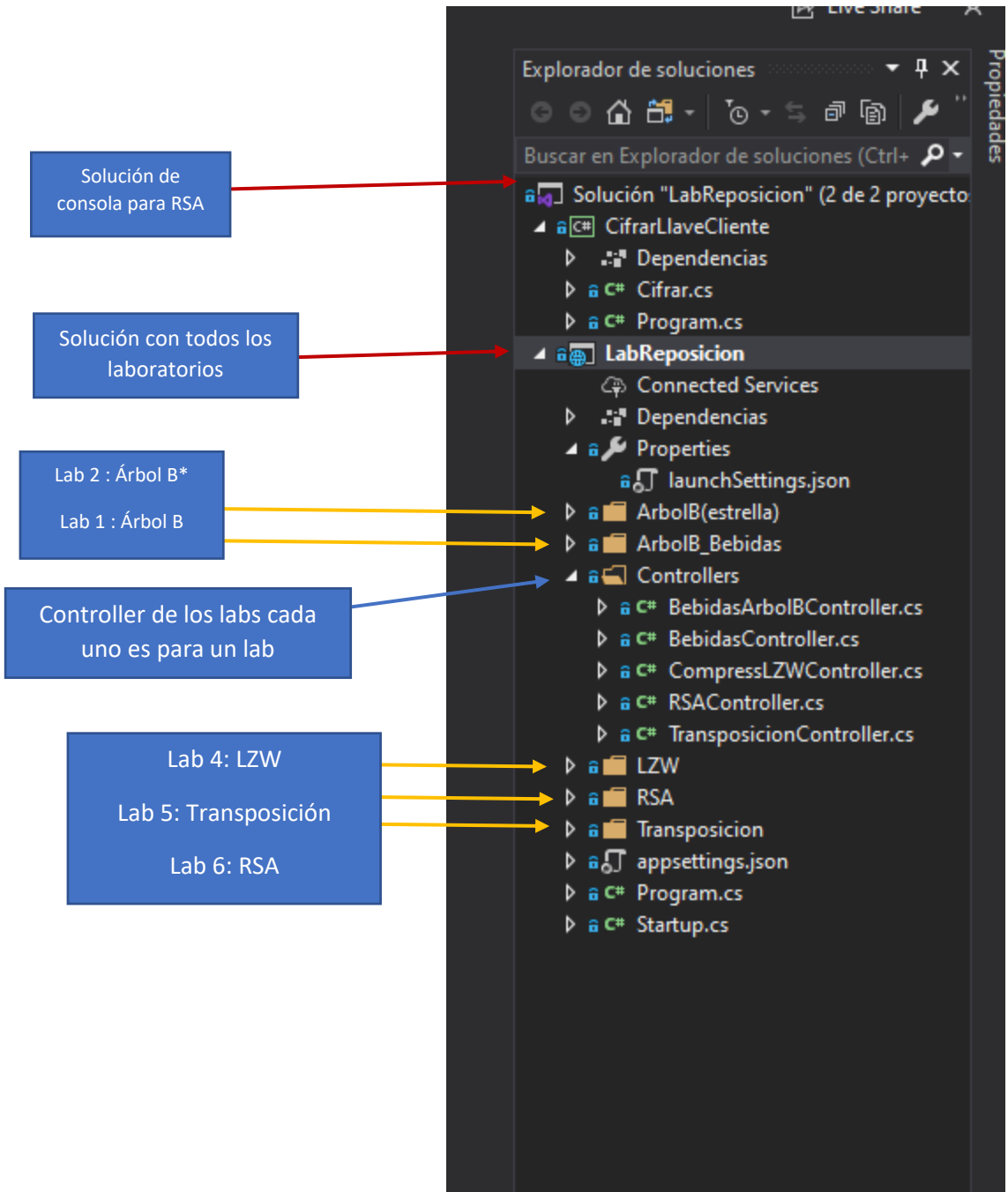
Laboratorio de reposición

Pablo Muralles 1113818

Santiago Bocel 1076818

Laboratorio Escogido

- 1.** Laboratorio 1: Árbol B
- 2.** Laboratorio 2: Árbol B*
- 3.** Laboratorio 4: LZW
- 4.** Laboratorio 5: Cifrados Transposición
- 5.** Laboratorio 6: RSA



Laboratorio 1: Árbol B

Ruta para Inserta (POST): <http://localhost:53138/api/insert>

Ejemplo para hacer una inserción:

```
{  
  "name": "Mirinda",  
  "flavor": "Naranja",  
  "inventory": 56,  
  "price": 4545,  
  "made": "CocaCola, S.A"  
}
```

Ruta para buscar (GET): <http://localhost:53138/api/buscar/{nombre a buscar}>

Se inserta en la ruta lo que se desea buscar y devuelve la búsqueda con todos los atributos de la bebida.

Ruta para obtener todos los registros (GET): <http://localhost:53138/api/registro>

Con esta ruta se obtienen todos los datos que contenga el árbol hasta ese momento.

Laboratorio 2: Árbol B*

Ruta para insertar (POST): <http://localhost:53138/Bebida/insertar>

Ejemplo:

```
{  
    "Nombre": "Coca",  
    "Sabor": "Cola",  
    "Volumen": 10000,  
    "Precio": 7.28,  
    "Casa_Productora": "CocaCola, S.A"  
}
```

Ruta para buscar una bebida (GET): <http://localhost:53138/Bebida/Buscar/{bebida a buscar}>

Se inserta en la ruta la bebida que se desea buscar y devuelve la búsqueda con todos los atributos de la bebida

Ruta para obtener los registros del árbol (GET): <http://localhost:53138/Bebida/registro>

Con esta ruta se obtienen todas las bebidas que contenga el árbol hasta ese momento.

Laboratorio 4: LZW

Ruta para comprimir un archivo (POST): <http://localhost:53138/compress/LZW/{nombre del archivo comprimido}>

Se manda el archivo que se desea comprimir por medio de postman en un from data en el body.

El archivo comprimido se guarda en una carpeta en la solución del proyecto llamada CompressLZW y el archivo comprimido tendrá el nombre que le haya mandado el usuario.

Ruta para descomprimir un archivo (POST): <http://localhost:53138/decompress/LZW/{Nombre del archivo descomprimido}>

Se manda el archivo que se desea descomprimir por medio de postman en un from data en el body.

El archivo descomprimido se guarda en una carpeta en la solución del proyecto llamada decompressLZW y el archivo descomprimido tendrá el nombre que le haya mandado el usuario.

Ruta para obtener el historial de las compresiones realizada (GET):
<http://localhost:53138/compression/LZW>

Devuelve el historial de todas las compresiones realizadas con sus atributos y cálculos.

Laboratorio 5: Cifrados Transposición

Ruta para cifrar un archivo (POST): <http://localhost:53138/cipher/{nombre del nuevo archivo cifrado que se va generar}>

Puede escoger entre estos 4: cesar, zigzag, Vertical, espiral. En nombre del cifrado debe de mandarse en la url tal y como se escribieron con anterioridad.

Las claves y los archivos se mandan en formando json.

Ejemplo para cifrar:

```
{  
  "fileName": "Nombre del archivo cifrado",  
  "carriles": ,  
  "path": " Ruta del archivo del a cifrar ",  
  "filas":  
}
```

Para zigzag se usan carriles como clave para vertical y espiral se usan filas.

Ruta para descifrar un archivo (POST): <http://localhost:53138/decipher/{nombre del nuevo archivo descifrado que se va generar}>

Puede escoger entre estos 4: cesar, zigzag, Vertical, espiral. En nombre del cifrado debe de mandarse en la url tal y como se escribieron con anterioridad. Y debe de mandar el nombre del cifrado con que se cifro el archivo sino no se va a descifrar correctamente.

Las claves y los archivos se mandan en formando json.

Ejemplo para descifrar:

```
{  
  "fileName": "Nombre del archivo descifrado",  
  "carriles": ,  
  "path": " Ruta del archivo del a cifrado ",  
  "filas":  
}
```

Para zigzag se usan carriles como clave para vertical y espiral se usan filas.

Laboratorio 6: RSA

Ruta para generar Llave Publica de RSA (GET): <http://localhost:53138/cipher/getPublicKey>

Se debe de mandar una llave en formato json para poder generar las llaves.

Ejemplo:

```
{  
  "key":23  
}
```

Va a retornar la llave publica para poder cifrar la llave del cliente.

Para poder Cifrar la llave del cliente se utiliza el programa de consola donde se mete la llave publica de RSA y se mete la llave con que se desea cifrar el texto con cesar. La llave del cliente cifrada con RSA se guarda en una en un txt en el debug del proyecto de consola.

Ruta para cifrar con cesar un archivo (POST): <http://localhost:53138/cipher/cesar2>

La contraseña para cesar, la ruta del archivo para cifrar y el nombre del archivo se mandan en formato json.

Ejemplo:

```
{  
  "RutaLlave": "Ruta de la contraseña cifrada generada por la consola",  
  "RutaArchivo": " Ruta del archivo que se desea cifrar ",  
  "NombreArchivo": "nombre del nuevo archivo cifrado"  
}
```

Ruta para descifrar con cesar un archivo (POST): <http://localhost:53138/decipher/cesar2>

La contraseña para cesar, la ruta del archivo para descifrar y el nombre del archivo descifrado se mandan en formato json.

Ejemplo:

```
{  
  "RutaLlave": "Ruta de la contraseña cifrada generada por la consola",  
  "RutaArchivo": " Ruta del archivo que se desea descifrar ",  
  "NombreArchivo": "nombre del nuevo archivo descifrado"  
}
```

Nota: A solo se puede descifrar mientras no se genere una nueva llave en el endpoint getpublickey.

