Criterio C. Desarrollo del producto

Programación orientada a objetos

Objetos

Dentro del programa se utiliza esta técnica para guardar las diferentes cualidades del objeto “Evento” y poder imprimirlas en un archivo de acceso o modificarlas. El método se instancia con la pantalla 3, donde se crea un evento, teniendo como parámetros formales lo obtenido de la página.

```
public class Evento {

    private boolean activo;
    private String fecha;
    private int horaDeInicio;
    private int horaDeFin;
    private String nombre;
    private Materia materia;
    private boolean tieneValor;
    private double calificacion;
    private double porcentaje;

    ArrayList<String> datos = new ArrayList<String>();
    ArrayList<Materia> materias = new ArrayList<Materia>();
    String linea;
```

La variable fecha esta guardada en string para que en la hora en que sea leida del documento o puesta por el usuario al crear un evento, siempre tenga el mismo formato numérico de DD-MM-YYYY (ej. 31-12-2019.) horaDeInicio y horaDeFin son int debido a que se trabaja con ellos a base de índices, 12 pm siendo 0, 12:30 uno y sucesivamente, para siempre poder distinguir cuál va antes. Las ultimas tres variables solo pueden ser cambiadas por la pantalla 4, donde se le pueda asignar una calificación al evento y promediar dentro de la materia que se aplica. Una pantalla donde se muestra esto es la pantalla 1, donde cada día muestra sus eventos.

Interfaz gráfica

En la primera pantalla se encuentra un calendario que se ajusta al día que empieza el mes, se va actualizando cada mes e ilumina con un color naranja el día actual.

Esto funciona con la biblioteca Calendar, donde declaras en que mes está y cuando empieza, después de eso va agregando un botón por cada día que lleva a la segunda pantalla ya instanciados como un arreglo al principio. Esta función da unos días antes y después del mes, por lo que se tiene asegurar que es el mes deseado. Cada vez que se quiera cambiar el mes, se cambia el calendario y el mes del arreglo meses al que se igualaba y luego se coloca cada día de la semana arriba. Para conseguir que las actividades aparecieran dentro del día, se busco cuales actividades coincidían en la fecha particular y se agregaban a un String que era luego añadido al botón junto con el número.

```
for(int week = 0; week < 6; week++) {
    for(int d = 0; d < 7; d++) {

        if(mesi.format(cal.getTime( )).equals(meses[mesMostrado])) {

            String frase = "<body><br>";

            for (int i = 0; i < eventosDelMes.size(); i++)
                if(eventosDelMes.get(i).getFecha().substring(0, 2).equals(day.format(cal.getTime( ))) &&
                    (Integer.parseInt(eventosDelMes.get(i).getFecha().substring(3, 5)) == (mesMostrado + 1)))
                    frase = frase + "-" + eventosDelMes.get(i).getNombre() + "\n";

            dias[a] = new JButton();
            dias[a].setText("<html>" + day.format(cal.getTime( )) + frase + "<html>");
            dias[a].setHorizontalAlignment(SwingConstants.LEFT);
            dias[a].setVerticalAlignment(SwingConstants.TOP);
            dias[a].setBounds(d*131+290, week*88+131, 130, 87);
            dias[a].setBackground(new Color(253, 127, 124));
            dias[a].setOpaque(true);
            dias[a].setFont(new Font("American Typewriter", 0, 18));
            dias[a].addActionListener(this);
            if((day.format(cal.getTime( )).equals(day.format(Calendar.getInstance().getTime()))) &&
                    (mesi.format(cal.getTime( )).equals(mesi.format(Calendar.getInstance().getTime()))))
                    dias[a].setForeground(new Color(253, 127, 124));
            add(dias[a], -1);
            dias[a].revalidate();
            dias[a].repaint();
            a++;
        }
        cal.add(Calendar.DATE,+1);
    }
}
```

Para cambiar de mes se utilizan las flechas en los costados del mes. Estos se logran cambiar con la misma biblioteca de Calendar. Después de que cada botón haya sido creado, se mueve un día más para poner el siguiente, quedando al final el siguiente mes. Cuando se usa una flecha, todos los botones de días se vuelven invisibles para luego crear el nuevo mes. Para cambiar el mes para atrás se ajusta el día al primero del mes, luego se le restan dos meses al calendario, mientras que la flecha siguiente solo empieza otra vez el ciclo de colocar los botones de días. LA variable de mes mostrado es el nombre del mes que cambia debido a las flechas, es un int indicando la posición de los meses en español en un array protegido.

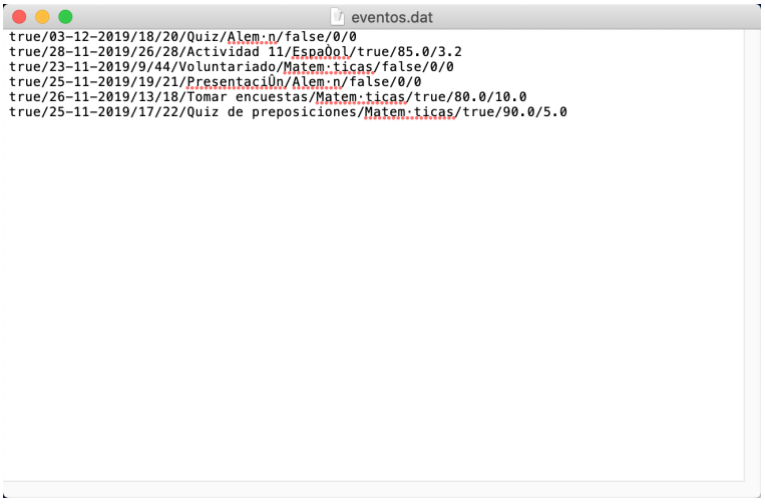
```
if(e.getSource() == flechaIzquierda) {
    for (int i = 0; i < a; i++) {
        dias[i].setVisible(false);
    }

    cal.add(Calendar.MONTH, -2);
    cal.set(Calendar.DATE,1);

    mesMostrado = mesMostrado - 1;
    if(mesMostrado == -1)
        mesMostrado = 11;
    frase1.setVisible(false);
    botonesDeDias();
}

if(e.getSource() == flechaDerecha) {
    for (int i = 0; i < a; i++) {
        dias[i].setVisible(false);
    }

    mesMostrado = mesMostrado + 1;
    if(mesMostrado == 12)
        mesMostrado = 0;
    frase1.setVisible(false);
    botonesDeDias();
}
```

Imprimir en un archivo de acceso

El archivo de acceso es usado ya que mi cliente es el único con acceso a la computadora, por lo que no necesita encriptarlo y que solo hay una computadora, no tiene que ser por internet. Esta se guarda en un orden especifico, primero si esta activo o no, para delimitar la búsqueda. Luego sigue la fecha, las horas donde empieza y termina y la clave. Después sigue el nombre de la materia y lo relacionado con las calificaciones., todo esto está escrito en bytes y dividido por slashes para garantizar una lectura y separación segura a la hora de recuperar los datos.

```java
public void imprimirEvento(RandomAccessFile archivo) {

    try {

        archivo.writeBytes(String.valueOf(isActivo()));
        archivo.writeBytes("/");

        archivo.writeBytes(getFecha());
        archivo.writeBytes("/");

        archivo.writeBytes(String.valueOf(getHoraDeInicio()));
        archivo.writeBytes("/");

        archivo.writeBytes(String.valueOf(getHoraDeFin()));
        archivo.writeBytes("/");

        archivo.writeBytes(getNombre());
        archivo.writeBytes("/");

        archivo.writeBytes(getMateria().getNombre());
        archivo.writeBytes("/");

        archivo.writeBytes(String.valueOf(isTieneValor()));
        archivo.writeBytes("/");

        if(getCalificacion() != 0)
            archivo.writeBytes(Double.toString(getCalificacion()));
        else
            archivo.writeBytes("0");
        archivo.writeBytes("/");

        if(getPorcentaje() != 0)
            archivo.writeBytes(Double.toString(getPorcentaje()));
        else
            archivo.writeBytes("0");
        archivo.writeBytes("\n");

    }
    catch(IOException f) {
        System.out.println("Archivo no encontrado Evento:imprimirEvento");
    }

}
```

Esto también se puede ver en el documento .dat donde se están imprimiendo los eventos.



Para leer estos documentos se utiliza un método donde cada línea es dividida por un slash. Cada línea es un dato por lo que separarla permite que pueda instanciarse un nuevo objeto con los

datos obtenidos. Todo esto es utilizando el método .readLine(). Para comprobar que esta trabajando correctamente, se checa el resultado dentro del programa, donde tiene que estar en un día especifico con sus indicaciones especificas.

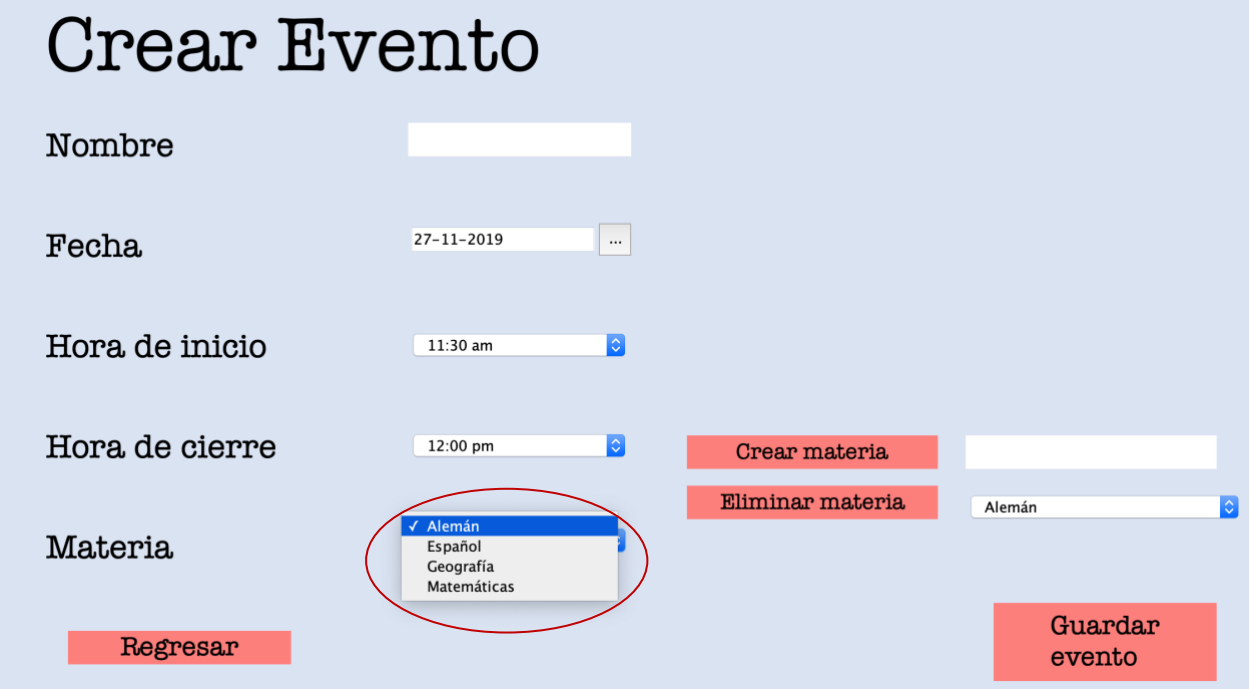
```
public void leerArchivoDeEventos() {
    try {

        RandomAccessFile archivo = new RandomAccessFile ("eventos.dat","rw");

        String linea = archivo.readLine();
        String[] eventoMomentario;

        if(linea != null)
            while(linea != null) {
                eventoMomentario = linea.split("/");

                Evento eventoTemporal = new Evento(Boolean.parseBoolean(eventoMomentario[0]), eventoMomentario[1],
                        Integer.parseInt(eventoMomentario[2]),Integer.parseInt(eventoMomentario[3]), eventoMomentario[4],
                        eventoMomentario[5], false, Double.parseDouble(eventoMomentario[7]), Double.parseDouble(eventoMomentario[8]));
                eventos.add(eventoTemporal);
                linea = archivo.readLine();
            }

        archivo.close();

    }
    catch(IOException e) {
        JOptionPane.showMessageDialog(null, "Error encontrando el archivo Pantalla1:leerArchivo");
    }
}
```

Ordenamiento burbuja de datos

Se utiliza el ordenamiento de datos para que cuando el usuario escoja una materia, estas estén ordenadas alfabéticamente y luego puedan ser buscadas por búsqueda binaria, que depende en el ordenamiento alfabético de estas. En este método se compara cada uno de los elementos y se hace una copia con otra variable para poner antes o después dependiendo cual es el mayor hasta pasar por todos los elementos y ordenar la lista.

```
public void ordenarLista(ArrayList<Materia> lista) {

    Materia temporal;

    for (int i = 0; i < lista.size(); i++)
        for (int j = i + 1; j < lista.size(); j++)
            if (lista.get(i).getNombre().compareTo(lista.get(j).getNombre()) > 0) {
                temporal = lista.get(i);
                lista.set(i, lista.get(j));
                lista.set(j, temporal);
            }
}
```

Dando el siguiente resultado



Búsqueda binaria

Se utilizó búsqueda binaria con el propósito de buscar el objeto de materia correspondiente al String obtenido por la opción elegida por el usuario, por lo que este método busca un objeto materia por su método de .get(int).getNombre() para poder compararlo con un string. Este es un método recursivo donde empieza con los datos de 0, tamaño del ArrayList y la palabra con la que se quiere encontrar el objeto materia, con cada pasada se define si la palabra es mayor o menor alfabéticamente al .getNombre() del objeto en medio de la lista y lo parte para delimitar la búsqueda y se vuelve a llamar.

```
public int buscarMateria(int b, int a, String materia) {

        if (a >= b) {
            int mid = b + (a - b) / 2;

            if (materia.compareTo(materias.get(mid).getNombre()) == 0)
                return mid;

            if (materia.compareTo(materias.get(mid).getNombre()) < 0)
                return buscarMateria(b, mid - 1, materia);

            return buscarMateria(mid + 1, a, materia);
        }

        return -1;

    }
```

Validación

Esta se utiliza al crear un nuevo evento, se utiliza validación para que este tenga todos los datos necesarios para otras pantallas antes de imprimir el evento y no haya un problema al leerlo posteriormente. Se utiliza un if y regresa un mensaje de como un usuario puede corregir el error, hasta que pase por todos, se utiliza el else para seguir con el programa. Esto se utiliza en diferentes partes del programa para poder comprobar que la información obtenido sea completa y no perturbe con un método que luego la necesite.

```
String nombre = paraEscribir1.getText();
String fecha = datePicker.getJFormattedTextField().getText();
int horaDeInicio = combo1.getSelectedIndex();
int horaDeFin = combo2.getSelectedIndex();
String materia = materias.get(combo3.getSelectedIndex());

if(paraEscribir1.getText().isEmpty())
    JOptionPane.showMessageDialog(this, "El evento requiere un nombre para ser creado");

else if (materias.get(0).equals("No hay materias para eliminar"))
    JOptionPane.showMessageDialog(this, "El evento requiere una nueva materia");

else if (horaDeFin <= horaDeInicio)
    JOptionPane.showMessageDialog(this, "El evento requiere que la hora de inicio sea antes de cuando termina");

else {

    try {

    RandomAccessFile archivo = new RandomAccessFile ("eventos.dat", "rw");
```

Conteo de palabras: 911 palabras

Bibliografía

- Aioobe. (10 de febrero de 2011). Preferred way of getting the selected item of a JComboBox. Recuperado de https://stackoverflow.com/questions/4962416/preferred-way-of-getting-the-selected-item-of-a-jcombobox
- Bisciak, T. (24 de enero de 2014). confirmation before press YES to exit program in Java. Recuperado de https://stackoverflow.com/questions/21330682/confirmation-before-press-yes-to-exit-program-in-java
- Casorzo, C. (s.f.). Estructuras de Datos Abstractas en Lenguaje Java. Recuperado de https://www.cec.uchile.cl/~luvasque/edo/java/manuales/Estructuras de Datos en Lenguaje Java (CCG).pdf
- Chitranayal. (s.f.). Bubble Sort. Recuperado de https://www.geeksforgeeks.org/bubble-sort/
- Eran. (13 de julio de 2015). How to do sorting in array list without using collections in Java. Recuperado de https://stackoverflow.com/questions/31377448/how-to-do-sorting-in-array-list-without-using-collections-in-java
- Fuente, Á. (12 de junio de 2018). La mosca tse-tsé que diezma el ganado africano. Recuperado de https://elpais.com/elpais/2018/06/06/planeta_futuro/1528271448_155734.html
- Ha Minh, N. (5 de julio de 2019). How to use JDatePicker to display calendar component. Recuperado de https://www.codejava.net/java-se/swing/how-to-use-jdatepickedeto-display-calendar-component
- Jenkov, J. (23 de junio de 2014). Java's java.util.TimeZone. Recuperado de http://tutorials.jenkov.com/java-date-time/java-util-timezone.html
- Jenkov, J. (2 de enero de 2019). Java List. Recuperado de http://tutorials.jenkov.com/java-collections/list.html
- Joy, H. (2011, December 14). Java - having buttons displaying arrows. Recuperado de https://stackoverflow.com/questions/8502068/java-having-buttons-displaying-arrows
- La Geekipedia De Ernesto. (18 de julio de 2017). Curso Java desde cero #38 | Interfaces gráficas (Swing - JCheckBox). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=AX8DynqTdsg&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=38
- La Geekipedia De Ernesto. (19 de julio de 2017,). Curso Java desde cero #39 | Botón para Aceptar términos y condiciones (JButton - JCheckBox). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=tBM8D2nj_u4&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=39
- La Geekipedia de Ernesto. (20 de julio de 2017). Curso Java desde cero #40 | Interfaces gráficas (Swing - JRadioButton). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=IP_UIGj6xmc&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=40

- La Geekipedia De Ernesto. (4 de julio de 2017). Curso Java desde cero #35 | Interfaces gráficas - Botón RGB. Recuperado de [Archivo de video] https://www.youtube.com/watch?v=7kUv991zh_o&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=35
- La Geekipedia de Ernesto. (6 de julio de 2017). Curso Java desde cero #36 | Programación de un menú con eventos (JMenuBar - JMenu - JMenuItem). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=sNFWxrCJb4s&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=36
- La Geekipedia De Ernesto. (7 de julio de 2017). Curso Java desde cero #37 | Programación de Submenus. Recuperado de [Archivo de video] https://www.youtube.com/watch?v=WsPQ7Z_8m9g&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=37
- La Geekipedia De Ernesto. (12 de junio de 2017). Curso Java desde cero #31 | Interfaces gráficas (Swing - JScrollPane). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=2Mob_1pizSw&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=31
- La Geekipedia De Ernesto. (26 de junio de 2017). Curso Java desde cero #32 | Como pasar texto de un JTextField a un JTextArea. Recuperado de [Archivo de video] https://www.youtube.com/watch?v=iyBH1_EoR9w&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=32
- La Geekipedia De Ernesto. (27 de junio de 2017). Curso Java desde cero #33 | Conversión de datos (Método Parse). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=g3gE5eQMYS8&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=33
- La Geekipedia De Ernesto. (28 de junio de 2017). Curso Java desde cero #34 | Interfaces gráficas (Swing - JComboBox). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=ZcTVwj4lHZA&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=34
- La Geekipedia de Ernesto. (6 de junio de 2017). Curso Java desde cero #29 | Interfaces gráficas (Swing - JTextField). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=Rs4gz77Ap0g&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=29
- La Geekipedia de Ernesto. (7 de junio de 2017). Curso Java desde cero #30 | Interfaces gráficas (Swing - JTextArea). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=-pUDnJX5A0A&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=30
- La Geekipedia de Ernesto. (16 de mayo de 2017). Curso Java desde cero #24 | Interfaces gráficas (Librería swing). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=E3_3bF_q64k&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=24
- La Geekipedia de Ernesto. (18 de mayo de 2017). Curso Java desde cero #25 | Interfaces gráficas (swing - JFrame). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=VOGula26mMM&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=25

- La Geekipedia de Ernesto. (19 de mayo de 2017). Curso Java desde cero #26 | Interfaces gráficas (swing - JLabel). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=2xLHMUdSPMU&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=26
- La Geekipedia De Ernesto. (30 de mayo de 2017). Curso Java desde cero #27 | Interfaces gráficas (Swing - JButton). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=6PMMn62pf78&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=27
- La Geekipedia De Ernesto. (31 de mayo de 2017). Curso Java desde cero #28 | Interfaces gráficas (Manejo de botones & etiquetas). Recuperado de [Archivo de video] https://www.youtube.com/watch?v=oJl8inUGzGY&list=PLyvsggKtwbLX9LrDnl1-K6QtYo7m0yXWB&index=28
- MadProgrammer. (7 de noviembre de 2014). How do I implement JDatePicker. Recuperado de https://stackoverflow.com/questions/26794698/how-do-i-implement-jdatepicker
- Oracle. (s.f.). How to Use Combo Boxes. Recuperado de https://docs.oracle.com/javase/tutorial/uiswing/components/combobox.html
- Prabhu, R. (16 de enero de 2020). Binary Search. Recuperado de https://www.geeksforgeeks.org/binary-search/
- Prabhu, R. (s.f.). Binary Search. Recuperado de https://www.geeksforgeeks.org/binary-search/
- Sahni, B. (s.f.). ArrayList in Java. Recuperado de https://www.geeksforgeeks.org/arraylist-in-java/
- ShivamKD. (18 de junio de 2018). Java Swing: JTable. Recuperado de https://www.geeksforgeeks.org/java-swing-jtable/
- Singh, C. (s.f.). Java Program to Sort Strings in an Alphabetical Order. Recuperado de https://beginnersbook.com/2018/10/java-program-to-sort-strings-in-an-alphabetical-order/
- Stitane, M. (5 de noviembre de 2015). How to extract and reset date from JDatePicker? Recuperado de https://stackoverflow.com/questions/30831283/how-to-extract-and-reset-date-from-jdatepicker
- Striver. (s.f.). ArrayList get(index) method in Java with examples. Recuperado de https://www.geeksforgeeks.org/arraylist-get-method-java-examples/