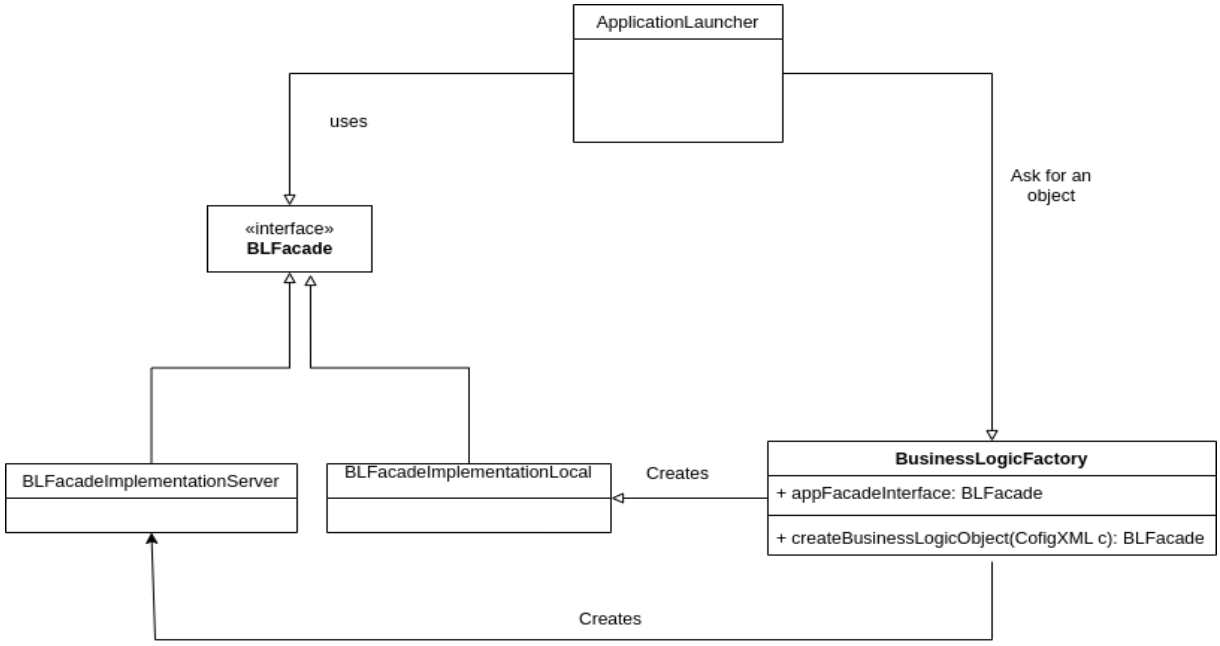
PATRONES

DOCUMENTACIÓN

Hasib Murib, Andrea Garcia

- **Patrón Factory Method**

  - ☐ **UML Diagram**



Para este ejercicio se ha utilizado el patrón simple factory que se encarga de crear objetos de BusinessLogic bien sea de manera local o de manera remota.

Como se puede observar en el diagrama, se ha creado la clase BusinessLogicFactory que utiliza el método estático de createBusinessLogicObject que recibe como parámetro el objeto ConfigXML. Después de comprobar que tipo de objeto le pide, procede a crearlo y finalmente devuelve el objeto.

☐ Código

Las partes del código que han sido modificadas están indicadas en color rojo.

```
package gui;

public class ApplicationLauncher {



	public static void main(String[] args) {

		ConfigXML c=ConfigXML.getInstance();

		System.out.println(c.getLocale());

		Locale.setDefault(new Locale(c.getLocale()));

		System.out.println("Locale: "+Locale.getDefault());

		MainGUI a=new MainGUI();
		a.setVisible(false);

		MainUserGUI b = new MainUserGUI();
		b.setVisible(true);


		try {

			BLFacade appFacadeInterface;

UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");

			appFacadeInterface =
BusinessLogicFactory.createBusinessLogicObject(c);

			MainGUI.setBussinessLogic(appFacadeInterface);


		}catch (Exception e) {
			a.jLabelSelectOption.setText("Error: "+e.toString());
			a.jLabelSelectOption.setForeground(Color.RED);
```

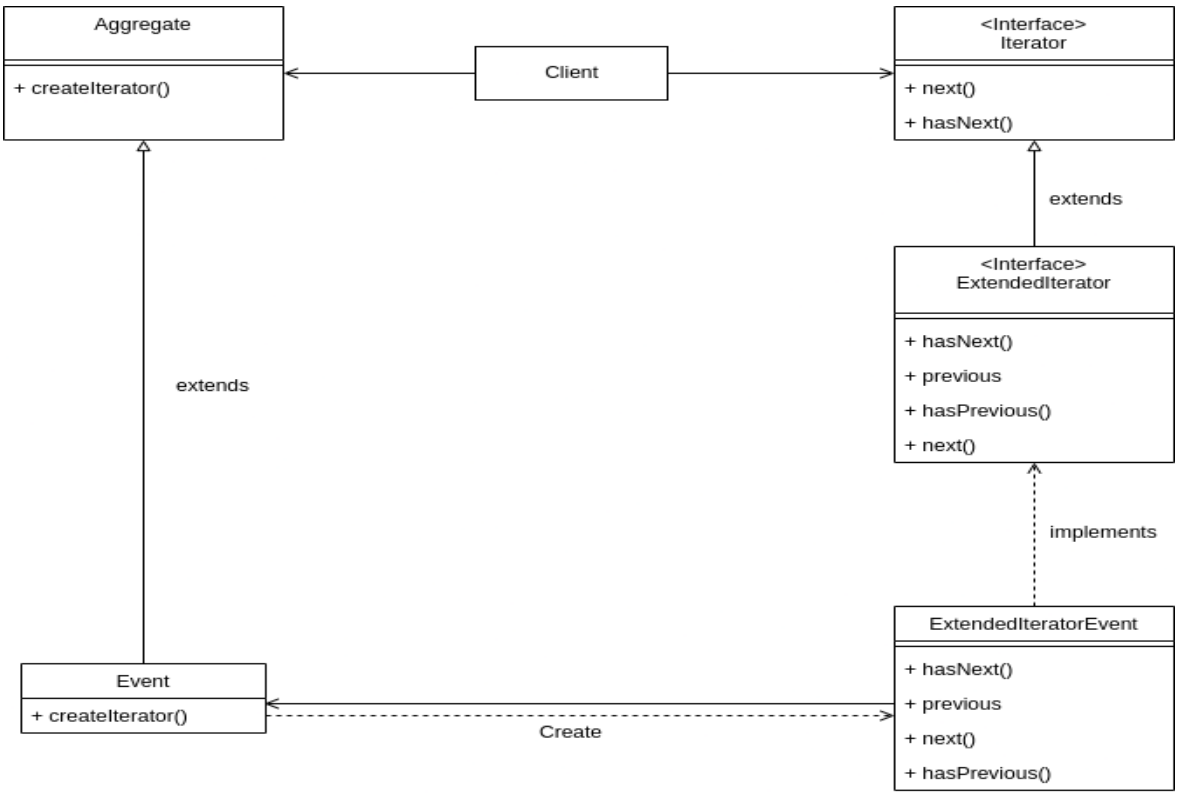
```
                    System.out.println("Error in ApplicationLauncher:
"+e.toString());
                }



        }

}
```

```
package businessLogic;

public class BusinessLogicFactory {

        static BLFacade appFacadeInterface;
        public BusinessLogicFactory() {

        }

        public static BLFacade createBusinessLogicObject(ConfigXML c) {

                try {
                        if (c.isBusinessLogicLocal()) {

                                DataAccess da= new
DataAccess(c.getDataBaseOpenMode().equals("initialize"));
                                appFacadeInterface=new
BLFacadeImplementation(da);
                        }

                        else {

                                String serviceName=
"http://"+c.getBusinessLogicNode() +":"+
c.getBusinessLogicPort()+"/ws/"+c.getBusinessLogicName()+"?wsdl";

                                URL url = new URL(serviceName);
                                QName qname = new
QName("http://businessLogic/", "BLFacadeImplementationService");
```

```
                        Service service = Service.create(url, qname);

                        appFacadeInterface =
service.getPort(BLFacade.class);

                    }

                } catch(Exception e) {
                    System.out.println("Error in ApplicationLauncher:
"+e.toString());
                }
                return appFacadeInterface;
            }
}
```

- **Patrón Iterator**

  - ☐ **UML Diagram**



Para este ejercicio como se ha indicado en el enunciado del laboratorio se ha creado un interfaz llamada “ExtendedIterator” que además de los métodos de interfaz de “Iterator” contiene métodos como “previous” y “hasPrevious”.

La clase que implementa dicha interfaz la hemos llamado ExtendedIteratorEvent.

☐ Código

```
package businessLogic;
import java.util.List;

import domain.Event;

public class ExtendedIteratorEvents implements ExtendedIterator<Object> {

	List<Event> eventos;
	int position;

	public ExtendedIteratorEvents(List<Event> events) {
		this.eventos = events;
		this.position = 0;
	}


	@Override
	public boolean hasNext() {

		return position < eventos.size();
	}

	@Override
	public Object next() {

		Event evento = eventos.get(position);
		position +=1;
		return evento;
	}

	@Override
	public Object previous() {

		if (position>0) {
```

```
                        Event evento = eventos.get(position-1);
                        position -=1;
                        return evento;

                }

                else
                        return null;
        }

        @Override
        public boolean hasPrevious() {
                return position-1 != -1;
        }

        @Override
        public void goFirst() {
                position=0;
        }

        @Override
        public void goLast() {
                position=eventos.size();

        }

}
```

DataAccess.java

```
public ExtendedIterator<Event> getEventsIterator(Date date) {
                List<Event> events = this.getEvents(date);
                ExtendedIterator eventos = new
ExtendedIteratorEvents(events);
                return eventos;
        }
```

Se ha considerado que es más eficiente llamar anteriormente al método getEvents(Date date) para obtener la lista de eventos. Posteriormente se crea solo un objeto de ExtendedIteratorEvents pasándole como parámetro dicha lista.

BLFacadeImplementation.java

```
@Override
```

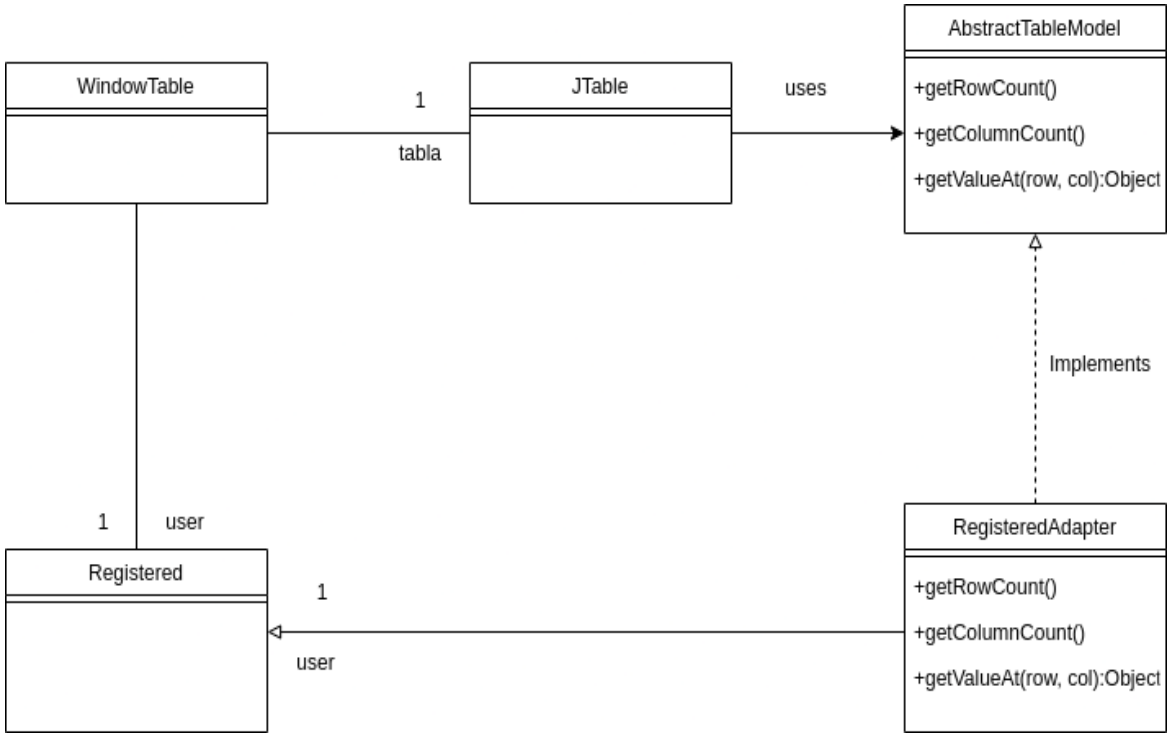
```
public ExtendedIterator<Event> getEventsIterator(Date date) {

        dbManager.open(false);
        ExtendedIterator<Event>
events=dbManager.getEventsIterator(date);
        dbManager.close();
        return events;
    }
```

☐ Ejecución

```
_____________________________________
Recorrido hacía atrás
27;Djokovic-Federer
24;Miami Heat-Chicago Bulls
23;Atlanta Hawks-Houston Rockets
22;LA Lakers-Phoenix Suns
10;Betis-Real Madrid
9;Real Sociedad-Levante
8;Girona-Leganes
7;Malaga-Valencia
6;Las Palmas-Sevilla
5;Espanol-Villareal
4;Alaves-Deportivo
3;Getafe-Celta
2;Eibar-Barcelona
1;Atletico-Athletic


_____________________________________
Recorrido hacía delante
1;Atletico-Athletic
2;Eibar-Barcelona
3;Getafe-Celta
4;Alaves-Deportivo
5;Espanol-Villareal
6;Las Palmas-Sevilla
7;Malaga-Valencia
8;Girona-Leganes
9;Real Sociedad-Levante
10;Betis-Real Madrid
22;LA Lakers-Phoenix Suns
23;Atlanta Hawks-Houston Rockets
24;Miami Heat-Chicago Bulls
27;Djokovic-Federer
```

- **Patrón Adapter**

  - ☐ **UML Diagram**



  - ☐ **Código**

```
package dataAccess;

public class RegisteredAdapter extends AbstractTableModel {


        private String[] columnNames = {"Event", "Question", "Event Date",
"Bet(€)"};

        Registered user;

        List<ApustuAnitza> apuestasMultiples;


        public RegisteredAdapter(Registered user) {
```

```
		this.user = user;
		this.apuestasMultiples=user.getApustuAnitzak();
	}


	@Override
	public int getRowCount() {
		if (this.apuestasMultiples==null)
			return 0;
		else
			return apuestasMultiples.size();
	}

	@Override
	public int getColumnCount() {
		return columnNames.length;
	}

	@Override
	public Object getValueAt(int row, int col) {

		Object temp = null;

		if (apuestasMultiples.get(row).getApustuak().size()>0) {

			if (col == 0) {
				temp =
apuestasMultiples.get(row).getApustuak().firstElement().getKuota().getQuestio
n().getEvent().getDescription();
			}
			else if (col == 1) {
				temp =
apuestasMultiples.get(row).getApustuak().firstElement().getKuota().getQuestio
n().getQuestion();

			}
			else if (col == 2) {
				temp =
apuestasMultiples.get(row).getApustuak().firstElement().getKuota().getQuestio
n().getEvent().getEventDate();
			}
			else if (col ==3) {
				temp = apuestasMultiples.get(row).getBalioa();
			}
			return temp;
		}
		else
			return null;
```

```
	}

	public String getColumnName(int col) {
		return columnNames[col];
	}


	public void imprimirApuestas() {
		for (ApustuAnitza a: apuestasMultiples) {
			if (a.getApustuak().size()>0) {

System.out.println(a.getApustuak().firstElement().getKuota().getQuestion().get
Event().getDescription()+",  "

+a.getApustuak().firstElement().getKuota().getQuestion().getQuestion()+", "

+a.getApustuak().firstElement().getKuota().getQuestion().getEvent().getEventD
ate()+", "
				+a.getBalioa());
				}
		}

	}

	public Class getColumnClass(int col) {
		if (col == 0 || col == 1) {
			return String.class;
		}
		else if (col == 2) {
			return Date.class;
		}
		else{
			return Double.class;
		}
	}

}
```

☐ Ejecución

Apuestas realizadas por andrea:

Event	Question	Event Date	Bet(€)
Atletico-Athletic	Who will win the match?	Dec 1, 2022	15
Atletico-Athletic	Who will score first?	Dec 17, 2022	17
Atletico-Athletic	Who will win the match?	Dec 17, 2022	9
Atletico-Athletic	Who will win the match?	Dec 17, 2022	225
Atletico-Athletic	Who will win the match?	Dec 1, 2022	678