

```

package main.java.com.wfdai.weatherforecastdai.main.weather;

import java.io.IOException; import javax.xml.bind.JAXBException; import
main.java.com.wfdai.weatherforecastdai.main.GestorErros;

/** * Fabrica de Objetos Weather @author daniel */ public class WeatherFac-
tory {

    GestorErros erros = new GestorErros();

    public WeatherFactory() {
    }

    public static WeatherInterface getWeather(String criteria) {
        if (criteria.equals("yahoo")) {
            return new WeatherYahoo();
        } else if (criteria.equals("OWM")) {
            return new OpenWeatherMaps();
        }
        return null;
    }

    /**
     * Recolhe os dados das API's externas e seleciona um com dados validos.
     *
     * @param localizacao String com a localização
     * @param weather Objecto com os dados meteorologicos recolhidos.
     * @throws javax.xml.bind.JAXBException
     * @throws java.io.IOException
     */
    public void setWeather(String localizacao, Weather weather) throws JAXBException, IOException {

        try {
            WeatherInterface weatherIn = WeatherFactory.getWeather("OWM");
            weatherIn.setWeather(localizacao, weather);

        } catch (NullPointerException e) {

            System.out.println("city not found");

            try {
                WeatherInterface weatherIn = WeatherFactory.getWeather("yahoo");
                weatherIn.setWeather(localizacao, weather);

            } catch (IOException ex) {
                erros.putErro("YW IOException: " + ex.getCause().toString());
            }
        }
    }
}

```

```
        } catch (JAXBException ex) {
            erros.putErro("YW JAXBException: " + ex.getCause().toString());
        }

    } catch (IOException ex) {
        erros.putErro("OWM IOException: " + ex.getCause().toString());
    } catch (JAXBException ex) {
        erros.putErro("OWM JAXBException: " + ex.getCause().toString());
    }
}
}
```