

A8 - Minería de Procesos

Análisis Exploratorio

1. ¿Cuántas ejecuciones del proceso contiene el log de eventos?

Son 6792 ejecuciones del proceso

```
Convert the received csv to a event log object

[10] import pm4py
      from pm4py.objects.conversion.log import converter as log_converter

      df_log_titles = pm4py.format_dataframe(df_csv, case_id="ID caso", activity_key="Nombre Actividad", timestamp_key="Fin")
      events_log = log_converter.apply(df_log_titles, variant=log_converter.Variants.TO_EVENT_LOG)

      events_log

[{'attributes': {'concept:name': '1'}, 'events': [{'ID caso': 1, 'Nombre Actividad': 'Alarma domiciliaria', 'Fin': Timestamp('2021-01-06 07:19:5')}]

Get the number of cases

len(events_log)

6792
```

2. ¿A qué periodo de tiempo corresponde?
3. ¿Cuáles son las actividades que se realizan más a menudo o menos a menudo?
Alarma Domiciliaria, Revisar Cámaras y Te atrapé son las que más se repiten, mientras que **Contactar seguridad ciudadana, Activar alarma comunitaria y Continuar disuasión y monitorear cámaras** son las que menos se repiten

```
Frequency of activities

df_events_log["Nombre Actividad"].value_counts()

Alarma domiciliaria      6792
Revisar cámaras          6792
Te atrapé                2701
Señal persona casa       2449
Registrar hecho fortuito  2101
Avisar hecho fortuito    2101
Registrar hecho no precisado 1990
Solicitar revisar alarma  1990
Clave digitada           1978
Concurrir a casa afectada  252
Revisar eventuales daños  252
Notificar llegada a central 252
Contactar dueño casa      252
Alertar móvil de seguridad 252
Dar mensajes disuasivos   252
Activar alarma casa       252
Registrar intruso detectado 252
Notificar fin operativo   252
Escalar alerta móvil      44
Contactar seguridad ciudadana 44
Activar alarma comunitaria 44
Continuar disuasión y monitorear cámaras 44
Name: Nombre Actividad, dtype: int64
```

4. ¿Con qué actividad se inicia el proceso? ¿Con qué actividades termina el proceso?
Todos los procesos inician con **Alarma domiciliaria**. Las actividades de término son

Señal persona casa, Clave digitada, Solicitar revisar alarma, Avisar hecho fortuito y Notificar fin operativo.

```
Starting and ending activities

[39] from pm4py.algo.filtering.log.start_activities import start_activities_filter
    from pm4py.algo.filtering.log.end_activities import end_activities_filter

    start_activities = start_activities_filter.get_start_activities(events_log)
    print("Starting Activities:")
    print(start_activities)

    end_activities = end_activities_filter.get_end_activities(events_log)
    print("Ending Activities:")
    print(end_activities)

Starting Activities:
{'Alarma domiciliaria': 6792}
Ending Activities:
{'Señal persona casa': 471, 'Clave digitada': 1978, 'Solicitar revisar alarma': 1990, 'Avisar hecho fortuito': 2101, 'Notificar fin operativo': 252}
```

5. ¿Cuántas variantes existen?

Existen 6 variantes

```
Existing variants

from pm4py.algo.filtering.log.variants import variants_filter

variants = variants_filter.get_variants(events_log)
len(variants)

6
```

6. ¿Cuáles son los ejecutores que participan realizando más actividades o menos actividades?

Sistema es el que realiza más actividades, en conjunto con **Nicolas** y **Francisca**.
Los que menos realizan actividades son **Javier**, **Cristóbal** y **Joaquin**

```
▼ Executor frequency

[29] df_events_log["Ejecutor"].value_counts()

Sistema      8770
Nicolas      3649
Francisca    3591
Thomas       3566
Bastian      3519
Martin       3505
Macarena     3478
Florencia    145
Natalia      140
Fernando     125
Tamara       115
Maria        110
Hernan       105
Fatima       95
Juan         95
Claudia      90
Cristobal    85
Joaquin      80
Javier       75
Name: Ejecutor, dtype: int64
```

7. ¿Cuál es el tiempo (mediana / promedio) que tarda el proceso en ejecutarse?
La mediana de lo que tarda el proceso en ejecutarse es: 481.0
La media de lo que tarda el proceso en ejecutarse es: 582.1647526501766

```
▼ Median/Mean of the process

from pm4py.statistics.traces.generic.log import case_statistics

all_case_durations = case_statistics.get_all_case_durations(
    events_log,
    parameters={
        case_statistics.Parameters.TIMESTAMP_KEY: "time:timestamp"
    },
)

import numpy as np

print(f"Median: {np.median(all_case_durations)}")
print(f"Mean: {np.mean(all_case_durations)}")
print(f"Standard Deviation: {np.std(all_case_durations)}")

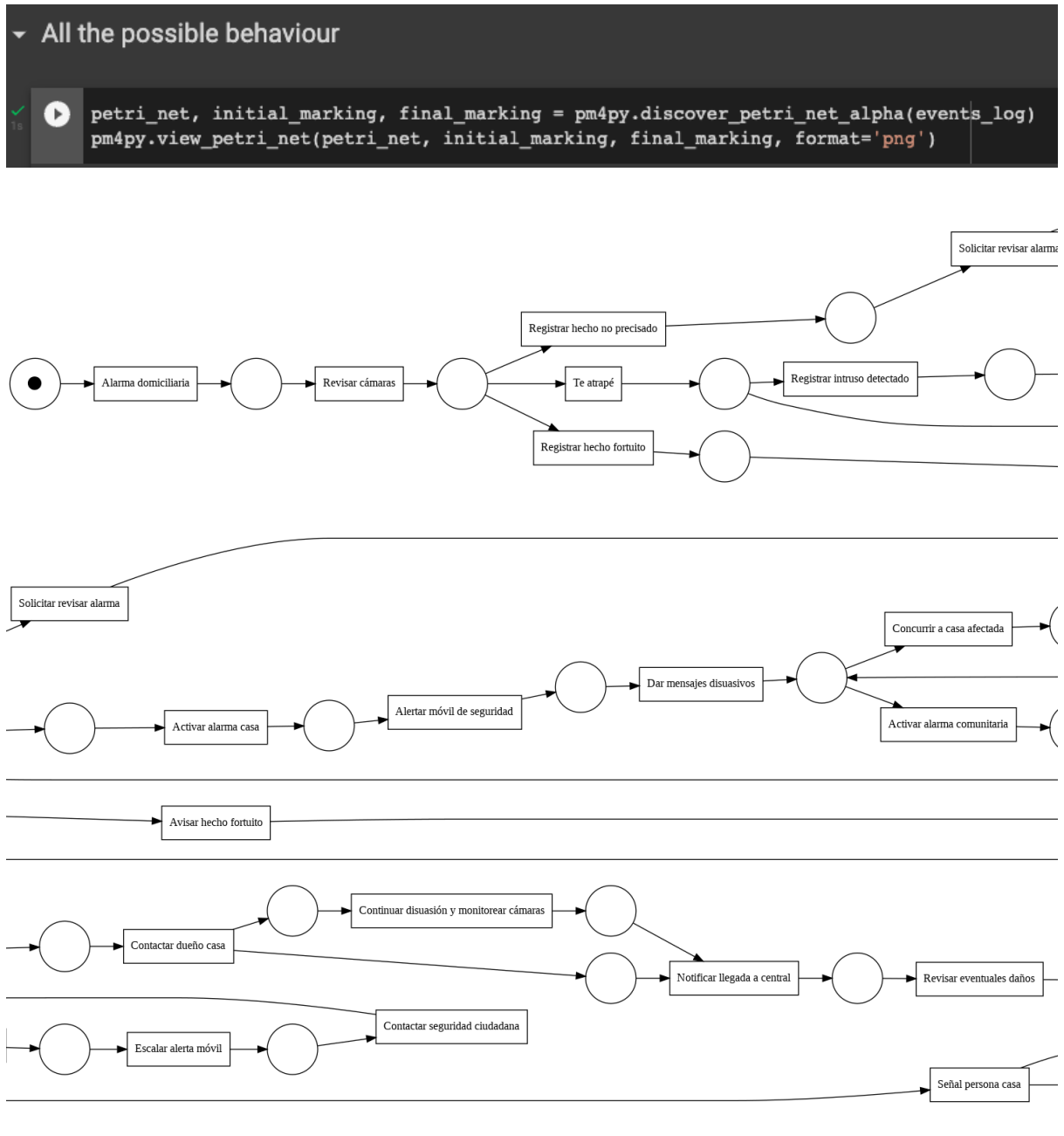
Median: 481.0
Mean: 582.1647526501766
Standard Deviation: 676.3723536293384
```

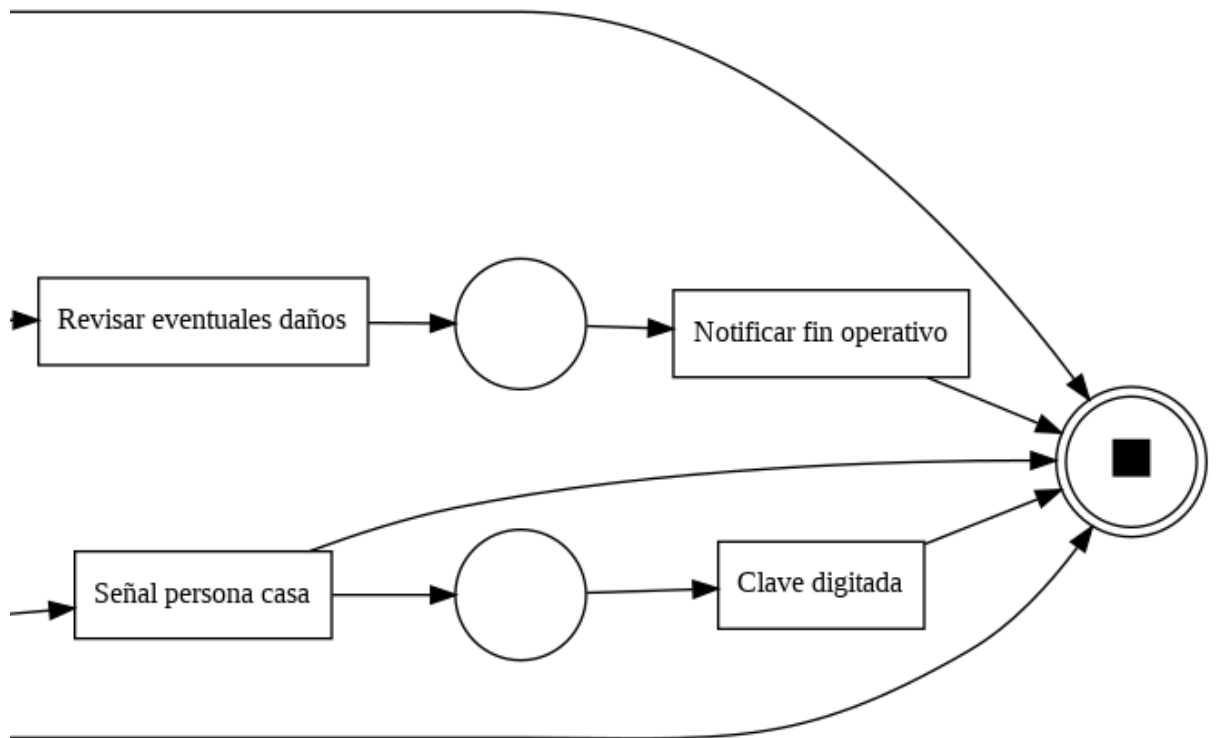
8. ¿Están todos los casos completos? Justifica tu respuesta.

Si están completos. Se puede decir esto debido a que la suma de la cantidad de actividades con las que finaliza el proceso es igual al número de actividades con las que se inicia, lo que se puede observar en el punto 4 de esta sección

Descubrimiento de Proceso

- Diagrama del proceso mostrando todo el comportamiento posible





- Diagrama de los casos que terminan sin recurrir la concurrencia de un movil

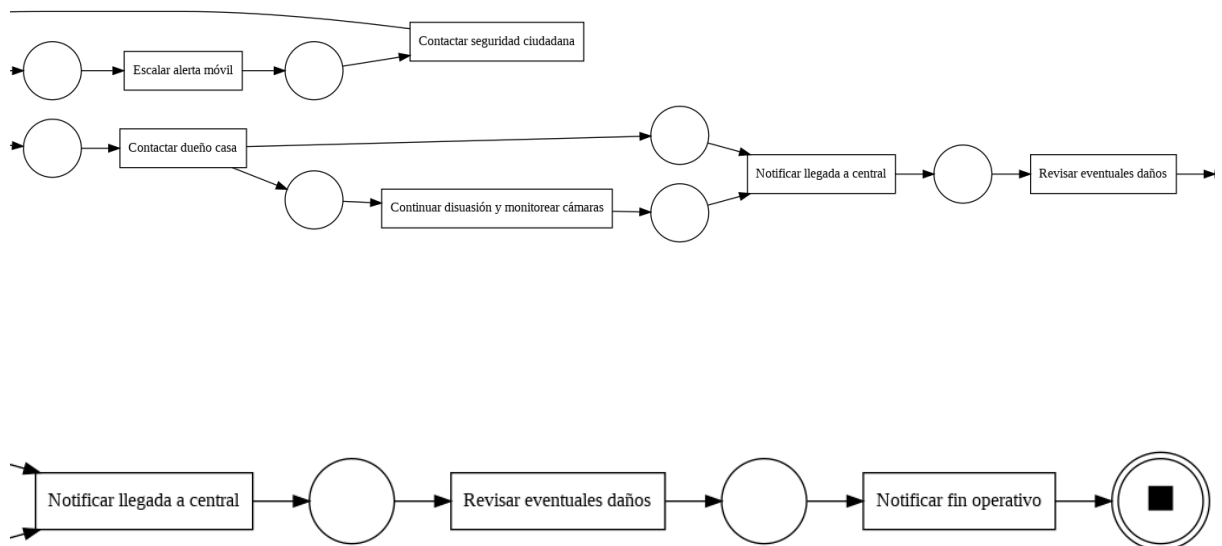
```

Case doesn't requires a car

from pm4py.algo.filtering.log.attributes import attributes_filter

requires_a_car_events_log = attributes_filter.apply(
    events_log,
    ["Alertar móvil de seguridad"],
    parameters={
        attributes_filter.Parameters.CASE_ID_KEY: "case:concept:name",
        attributes_filter.Parameters.ATTRIBUTE_KEY: "concept:name",
        attributes_filter.Parameters.POSITIVE: False,
    },
)

requires_a_car_petri_net, requires_a_car_initial_marking, requires_a_car_final_marking = pm4py.discover_petri_net_alpha(requires_a_car_events_log)
pm4py.view_petri_net(
    requires_a_car_petri_net,
    requires_a_car_initial_marking,
    requires_a_car_final_marking,
    format='png'
)
  
```

Si, el resultado es el esperado. Esto debido a que, en el caso de que no se requiera un móvil, se observa que las opciones de lo que puede haber pasado son mayores, esto porque implica que no hubo una emergencia real, si no que fue una activación accidental.

Mientras que, en el caso de que si se requiera uno, entonces aquello que activó la alarma se considera como una amenaza real, por lo que el camino a seguir es más lineal, ya que se debe seguir el protocolo que la empresa definió para hacer en estos casos.

Filtrar log de eventos

Tipo de caso	Frecuencia Log 1
Hecho fortuito	2101
Hecho no precisado	1990
Persona de la casa	2449
Intruso - no se requiere alerta comunitaria	208
Intruso - se requiere alerta comunitaria	44
Total	6792

```
[19] # Hecho fortuito
      fortuitous_event_events_log = attributes_filter.apply(
          events_log,
          ["Registrar hecho fortuito"],
          parameters={
              attributes_filter.Parameters.ATTRIBUTE_KEY: "concept:name",
              attributes_filter.Parameters.POSITIVE: True,
          },
      )

      len(fortuitous_event_events_log)
```

2101

```
▶ # Non especific event
    non_specific_event_events_log = attributes_filter.apply(
        events_log,
        ["Registrar hecho no precisado"],
        parameters={
            attributes_filter.Parameters.ATTRIBUTE_KEY: "concept:name",
            attributes_filter.Parameters.POSITIVE: True,
        },
    )

    len(non_specific_event_events_log)
```

1990

```
▶ # Person from the house
    from_the_house_event_events_log = attributes_filter.apply(
        events_log,
        ["Señal persona casa"],
        parameters={
            attributes_filter.Parameters.ATTRIBUTE_KEY: "concept:name",
            attributes_filter.Parameters.POSITIVE: True,
        },
    )

    len(from_the_house_event_events_log)
```

2449


```

intruder_event_events_log = attributes_filter.apply(
    events_log,
    ["Registrar intruso detectado"],
    parameters={
        attributes_filter.Parameters.ATTRIBUTE_KEY: "concept:name",
        attributes_filter.Parameters.POSITIVE: True,
    },
)

intruder_not_alarm_event_events_log = attributes_filter.apply(
    intruder_event_events_log,
    ["Activar alarma comunitaria"],
    parameters={
        attributes_filter.Parameters.CASE_ID_KEY: "case:concept:name",
        attributes_filter.Parameters.ATTRIBUTE_KEY: "concept:name",
        attributes_filter.Parameters.POSITIVE: False,
    },
)

len(intruder_not_alarm_event_events_log)

```

208

```

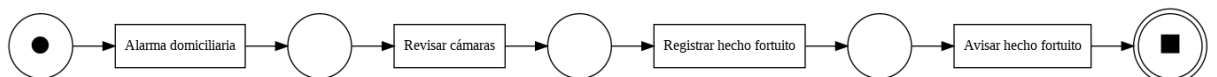
# Intruder && com alarm required
intruder_alarm_event_events_log = attributes_filter.apply(
    events_log,
    ["Activar alarma comunitaria"],
    parameters={
        attributes_filter.Parameters.ATTRIBUTE_KEY: "concept:name",
        attributes_filter.Parameters.POSITIVE: True,
    },
)

len(intruder_alarm_event_events_log)

```

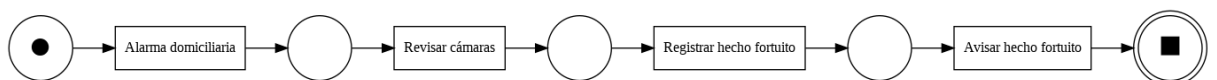
44

- Hecho fortuito



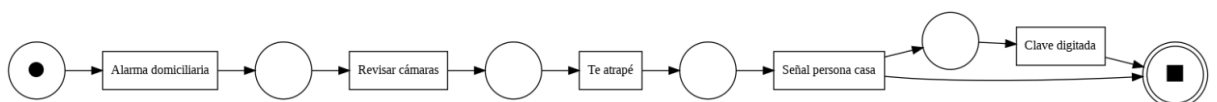
El proceso se ejecuta de acuerdo a lo esperado, donde los ejecutores ahora son personas y el sistema. Representa un 30,9% de los casos totales

- Hecho no precisado



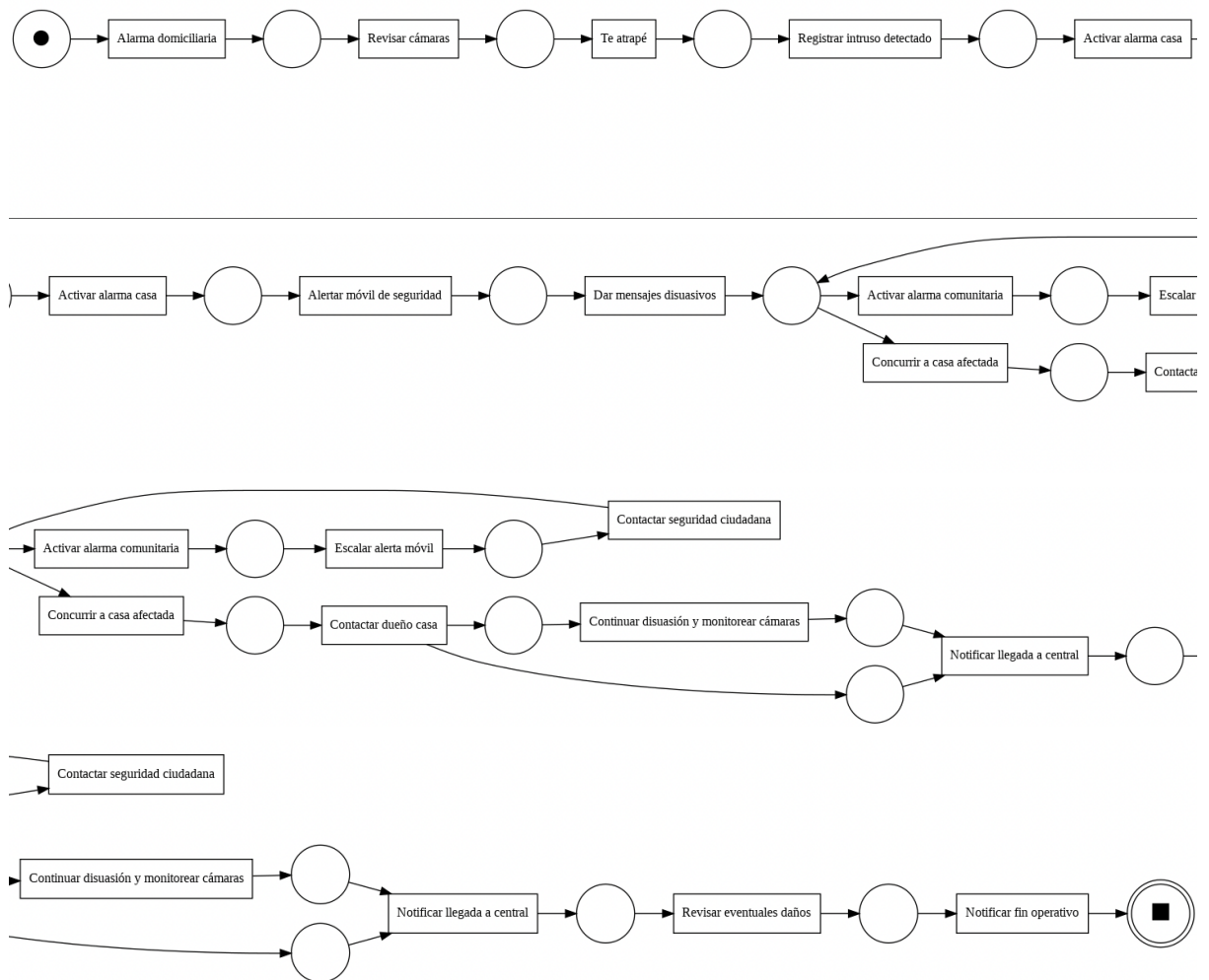
Representa un 29,2% de los casos totales

- Persona de la casa



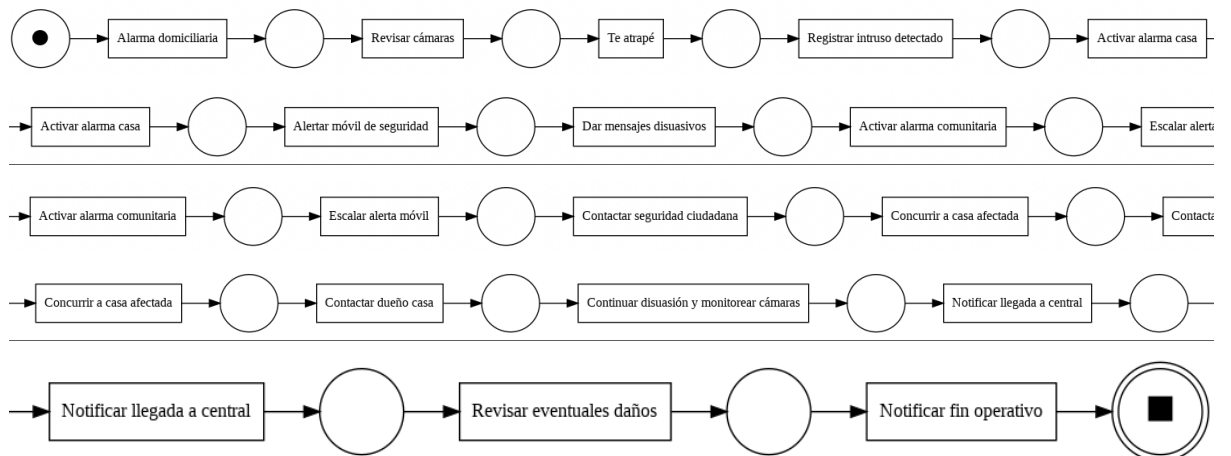
Representa un 36% de los casos totales

- Intruso - no se requiere alerta comunitaria



Representa un 3% de los casos totales

- Intruso - se requiere alerta comunitaria



Representa un 0,6% de los casos totales