

```

public EntradaVIP buscarEntrada(String nick, String plataforma) {
    Iterator<Influencer> it = registro.keySet().iterator();
    EntradaVIP entrada = null;
    while (it.hasNext()) {
        Influencer inf = it.next();
        if (inf.getNick().equals(nick) && inf.getPlataforma().equals(plataforma)) {
            entrada = registro.get(inf);
            break;
        }
    }

    return entrada;
}

```

```

public EntradaVIP buscarEntrada(String nick, String plataforma) {
    Iterator<Influencer> it = registro.keySet().iterator();
    EntradaVIP entrada = null;
    while (it.hasNext()) {
        Influencer inf = it.next();
        if (inf.getNick().equals(nick) && inf.getPlataforma().equals(plataforma)) {
            entrada = registro.get(inf);
            break;
        }
    }

    return entrada;
}

```

```

public boolean confirmarAsistencia(String nick, String plataforma) {
    EntradaVIP entrada = buscarEntrada(nick, plataforma);
    boolean asistenciaConfirmada = false;

    if (entrada != null) {
        entrada.confirmar();
        asistenciaConfirmada = true;
    }

    return asistenciaConfirmada;
}

```

```

public boolean cancelarAsistencia(String nick, String plataforma) {
    EntradaVIP entrada = buscarEntrada(nick, plataforma);
    boolean asistenciaCancelada = false;

    if (entrada != null) {
        entrada.cancelar();
        asistenciaCancelada = true;
    }

    return asistenciaCancelada;
}

```

```

public List<Influencer> getInfluencersPorEstado(EstadoEntrada estado) {
    List<Influencer> lista = new ArrayList<>();

    for (Map.Entry<Influencer, EntradaVIP> entry : registro.entrySet()) {
        if (entry.getValue() != null)
            if (entry.getValue().getEstado() == estado)
                lista.add(entry.getKey());
    }

    return lista;
}

```

```
public List<Influencer> obtenerTop3Influencers() {  
    // lista de influencers  
    List<Influencer> listaInfluencers = new ArrayList<>(registro.keySet());  
  
    //ordenar por número de seguidores (especificado en Comparable)  
    Collections.sort(listaInfluencers);  
    List<Influencer> top3 = new ArrayList<>();  
  
    // devolvemos los 3 primeros  
    for (int i = 0; i < 3 && i < listaInfluencers.size(); i++) {  
        Influencer inf = listaInfluencers.get(i);  
        top3.add(inf);  
    }  
  
    return top3;  
}
```