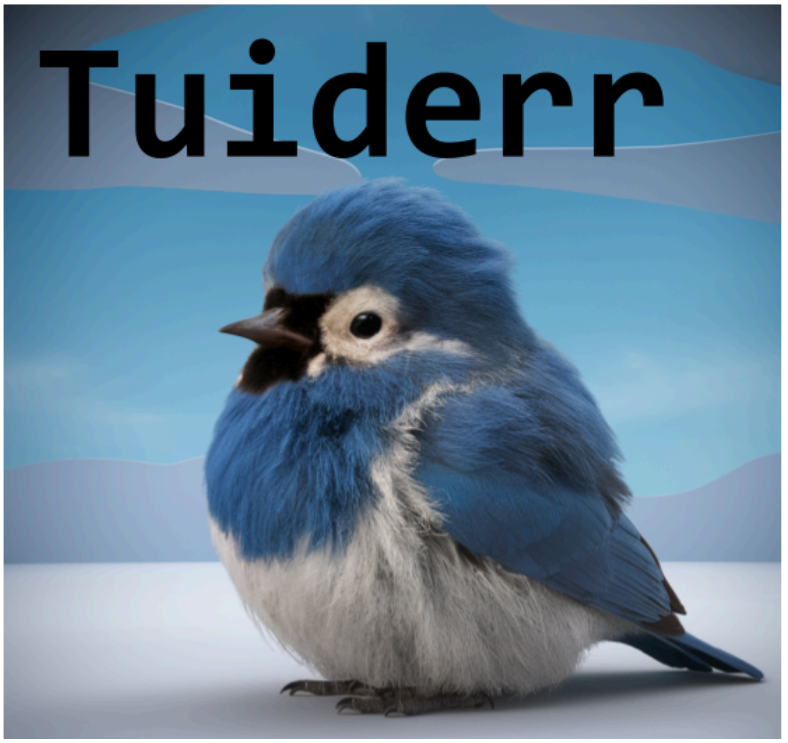
Tuiderr, la red social del *pajadito*, es aclamada por todo el mundo. Conecta a muchísimas personas (en su mayoría tóxicas) y permite mucha difusión de temas importantísimos... y también otros no tanto.



En Tuiderr, existen los Posts, que pueden ser formados de dos maneras:

1) Un Tuid, formado por el nombre de usuario del autor y el contenido.
2) Un ReTuid, formado por el nombre de usuario del retuideador y el post retuideado en sí mismo.

El contenido de un Tuid es una lista de las palabras.

De cada Usuario, además de sus Posts, se sabe su nombre de usuario y los Posts en proceso de aprobación.

```
% publicacion(Fecha, Post).
publicacion(fecha(25, 01, 2013), tuid(coirotomas, ["es", "mi", "cumple", "arrodillense"])).
publicacion(fecha(26, 01, 2014), tuid(fanDeTusk, ["paga", "Tuiderr", "en", "vez", "de", "quejarte", "queridito"])).
publicacion(fecha(06, 07, 2014), tuid(matifreyre, ["que", "bueno", "Les", "Luthiers"])).
publicacion(fecha(25, 12, 2014),
   tuid(melonTusk, ["ser", "millonario", "es", "más", "difícil", "que", "ser", "pobre"])).
publicacion(fecha(23, 07, 2023), tuid(melonTusk, ["ahora", "se", "llama", "Z"])).
publicacion(fecha(18, 05, 2024), tuid(alumnoFrustrado, ["pdep", "hace", "los", "peores", "parciales"])).
publicacion(fecha(05, 06, 2024),
   retuid(fanDeTusk, tuid(melonTusk, ["ser", "millonario", "es", "más", "difícil", "que", "ser", "pobre"]))).
publicacion(fecha(18, 05, 2024),
   retuid(alumno2, tuid(alumnoFrustrado, ["pdep", "hace", "los", "peores", "parciales"]))).
publicacion(fecha(23, 06, 2024), tuid(bornoPot, ["para", "mas", "contenido", "gatito", "enBio"])).
publicacion(fecha(23, 06, 2024), tuid(bornoPot, ["para", "menos", "contenido", "gatito", "enBio"])).
publicacion(fecha(23, 06, 2024), tuid(bornoPot, ["para", "otro", "contenido", "gatito", "enBio"])).
publicacion(fecha(23, 06, 2024), tuid(bornoPot, ["para", "diferente", "contenido", "gatito", "enBio"])).
publicacion(fecha(23, 06, 2024), tuid(bornoPot, ["para", "parecido", "contenido", "gatito", "enBio"])).
publicacion(fecha(23, 06, 2024), tuid(bornoPot, ["para", "distinto", "contenido", "gatito", "enBio"])).
publicacion(fecha(23, 06, 2024), tuid(coirotomas, ["compren", "bitcoin", "es", "el", "futuro"])).
publicacion(fecha(27, 06, 2024), tuid(estafador238, ["cryptos", "gratis", "en", "link", "pongan", "datos"])).
publicacion(fecha(06, 07, 2024), retuid(matifreyre, tuid(matifreyre, ["que", "bueno", "Les", "Luthiers"]))).

% paraAprobar(Fecha, Post).
paraAprobar(fecha(20, 03, 2023), tuid(leocesario, ["miren", "este", "lechoncito"])).
paraAprobar(fecha(25, 01, 2024), tuid(coirotomas, ["cumpli", "30", "me", "duele", "la", "espalda"])).
paraAprobar(fecha(17, 11, 2024),
   retuid(leocesario,  tuid(coirotomas, ["es", "mi", "cumple", "arrodillense"]))).
paraAprobar(fecha(23, 06, 2024), tuid(bornoPot, ["gatito", "enBio", "clickea", "el", "link"])).
```

Nos llega una solicitud desde Producto para mejorar algunas partes del sistema, para mejorar las interacciones de usuario y quitar cosas que los hacen enojar. Nos pidieron que modelemos las cosas en Prolog, porque quieren que parezca que todo esto lo hace una Inteligencia Artificial (PdePIA).

Nota: Recuerden que existen los predicados nth0/3 y nth1/3.

Aclaración: Cuando se hace referencia a una “publicación”, estamos hablando de un Post aprobado. En cambio, cuando hablamos de Post, puede ser aprobado (publicado) o en proceso (para aprobar).

1) Hacer **publicacionDe/2**, que relacione a un usuario con el contenido de una de sus publicaciones.
2) Implementar diferentes predicados de revisión de contenido:
    a) **contieneFrase/3** que relaciona dos palabras con un Post si las dos palabras están escritas en forma consecutiva en el contenido de ese Post.
    No necesita ser inversible para las dos palabras.
    b) **esDeBot/1**, que se verifica para cualquier Post que cumpla al menos una de las siguientes condiciones:
        i) Contiene "gatito enBio" en cualquier lugar del contenido.
        ii) No es de este año.
        iii) Es un ReTuid de un Post anterior a 2015.
3) Hacer **verdaderoAutor/2**, que relaciona un Post con su autor verdadero, siendo que, en los casos de los ReTuids, el verdadero autor es el que publicó el Tuid original.
4) Implementar **postsParaPublicar/2**, que relaciona un usuario y sus Posts en proceso de aprobación que no son de bot.
5) Necesitamos saber si un usuario debe ser **favorecido/1** por el algoritmo. Esto sucede sólo cuando, en sus Posts, no tiene ReTuids (él/ella no retuidea, es “100% ORISHINAL”), y además nunca incluyó la frase "cryptos gratis".
6) **esteEsWallE/1**, que se cumple para un usuario que tiene altas chances de ser un bot. Estos son los usuarios para los que todos sus Posts son de bot, o si existe un día en el que hizo más de 5 Posts.
7) Implementar **elMejorTimeline/2**, que relaciona un número de publicaciones pedidas, y una lista con esa cantidad de posts (sin repetir), que pueden ser:
    a) Posts de usuarios favorecidos.
    b) Posts donde el verdadero autor sea "melonTusk".
    c) Posts que contengan la frase "paga Tuiderr".
    No necesita ser inversible para la cantidad.

Tests:

```
:- include(tuiderr).

:- begin_tests(punto1_publicacionDe).
test(contenidoDeTuidDirecto, nondet):-
    publicacionDe(matifreyre, ["que", "bueno", "Les", "Luthiers"]).
test(contenidoDeReTuid, nondet):-
    publicacionDe(alumno2, ["pdep", "hace", "los", "peores", "parciales"]).
:- end_tests(punto1_publicacionDe).

:- begin_tests(punto2a_contieneFrase).
test(contieneFraseTuid, nondet):-
    contieneFrase("compren", "bitcoin", tuid(coirotomas, ["compren", "bitcoin", "es",
"el", "futuro"])).
test(noContieneFraseTuid, fail):-
    contieneFrase("compren", "futuro", tuid(coirotomas, ["compren", "bitcoin", "es",
"el", "futuro"])).
:- end_tests(punto2a_contieneFrase).

:- begin_tests(punto2b_esDeBot).
test(botDeGatito, nondet):-
    esDeBot(tuid(bornoPot, ["gatito", "enBio", "clickea", "el", "link"])).
test(botDeOtroAnio, nondet):-
    esDeBot(tuid(melonTusk, ["ahora", "se", "llama", "Z"])).
test(botRepiteViejo, nondet):-
    esDeBot(retuid(matifreyre, tuid(matifreyre, ["que", "bueno", "Les", "Luthiers"]))).
:- end_tests(punto2b_esDeBot).

:- begin_tests(punto3_verdaderoAutor).
test(verdaderoAutorDeTuid, nondet):-
    verdaderoAutor(alumnoFrustrado, tuid(alumnoFrustrado, ["pdep", "hace", "los",
"peores", "parciales"])).
test(verdaderoAutorDeReTuid, nondet):-
    verdaderoAutor(melonTusk, retuid(fanDeTusk, tuid(melonTusk, ["ser", "millonario",
"es", "más", "difícil", "que", "ser", "pobre"]))).
:- end_tests(punto3_verdaderoAutor).

:- begin_tests(punto4_postsParaPublicar).
test(tuidsParaPublicarSonTodosDeBot, nondet):-
    postsParaPublicar(leocesario, []).
test(tuidsParaPublicarSinBot, nondet):-
    postsParaPublicar(coirotomas, [tuid(coirotomas, ["cumplí", "30", "me", "duele", "la",
"espalda"])]).
:- end_tests(punto4_postsParaPublicar).

:- begin_tests(punto5_favorecido).
test(favorecido, nondet):-
    favorecido(coirotomas).
```

```
test(noFavorecidoPorReTuid, fail):-
    favorecido(matifreyre).
test(noFavorecidoPorFrase, fail):-
    favorecido(estafador238).
:- end_tests(punto5_favorecido).

:- begin_tests(punto6_esteEsWallE).
test(leocesarioEsWallEPorTenerPosteosDeBot, nondet):-
    esteEsWallE(leocesario).
test(bornoPotEsWallEPorTenerMuchosPosteosEnMismoDia, nondet):-
    esteEsWallE(bornoPot).
test(coirotomasNoEsWallE, fail):-
    esteEsWallE(coirotomas).
:- end_tests(punto6_esteEsWallE).

:- begin_tests(punto7_elMejorTimeline).
test(timelineConTodosLosCasosEnOrdenArbitrario, nondet):-
    elMejorTimeline(3,
        [tuid(coirotomas, ["compren", "bitcoin", "es", "el", "futuro"]),
         tuid(melonTusk, ["ahora", "se", "llama", "Z"]),
         tuid(fanDeTusk, ["paga", "Tuiderr", "en", "vez", "de", "quejarte",
"queridito"])]).
test(timelineConTodosLosCasosEnOtroOrdenArbitrario, nondet):-
    elMejorTimeline(3,
        [tuid(melonTusk, ["ahora", "se", "llama", "Z"]),
         tuid(fanDeTusk, ["paga", "Tuiderr", "en", "vez", "de", "quejarte",
"queridito"]),
         tuid(coirotomas, ["compren", "bitcoin", "es", "el", "futuro"])]).
:- end_tests(punto7_elMejorTimeline).
```