Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
CI-3391 - Laboratorio de Bases de Datos
Abril - Julio 2024
Profesor: Pedro Pérez

Integrantes:

Ana Shek	19-10096
Jhonaiker Blanco	18-10784
Junior Lara	17-10303
Laura Parilli	17-10778

TINDER PARA VIEJOS EGRESADOS v1.0 - 2da ENTREGA

1. Modelo Relacional

Enlace directo al diagrama ERE respectivo aquí.

- $CUENTA(\underline{Id\_cuenta}, Nombre, Apellido, Fecha\_nacimiento, Fecha\_creacion, Email, Contrasena, Telefono, Idioma, Notificaciones, Tema)$
- $PAGO(\underline{Id\_pago}, Numero\_factura, Estado, Metodo, Monto, Fecha, Documento\_factura)$
- $TARJETA(\underline{Digitos\_tarjeta}, Nombre\_titular, Fecha\_caducidad, Codigo\_CVV, Tipo)$
- $TIER(\underline{Nombre\_Tier})$
- $PERMISO(\underline{Nombre\_Permiso}, Descripcion\_Permiso)$
- $CHAT(\underline{Id\_chat})$
- $INSTITUCION(\underline{Dominio}, Nombre, Tipo, Anio\_fundacion, Latitud, Longitud)$
- $EMPRESA(\underline{Id\_empresa}, Nombre\_empresa, url)$
- $PERFIL(\underline{\overline{Id\_cuenta}^{Cuenta}}, Sexo, Descripcion, Verificado, Latitud, Longitud)$

  NOTA: Por las reglas de traducción del modelo ERE a relacional para entidades débiles, se tiene que colocar como PK a id_cuenta junto con la clave parcial id_perfil. Sin embargo, se coloca solamente id_cuenta como PK ya que cada cuenta tiene exactamente un perfil y por lo tanto basta con solo identificar el perfil con id_cuenta.

- $PREFERENCIAS(\overline{\underline{Id\_cuenta}}^{Cuenta}, Estudio, Distancia\_maxima, Min\_edad, Max\_edad, Latitud\_origen, Longitud\_origen)$

  NOTA: Lo mismo que en la nota anterior, el PK debería ser la combinación de id_cuenta, id_perfil, id_pref, pero como cada perfil solo puede estar asociado a una instancia de preferencias y cada cuenta tiene exactamente un solo perfil, es suficiente identificar las preferencias por id_cuenta.

- $MENSAJE(\overline{\underline{Id\_chat}}^{Chat}, Numero\_msj, Id\_remitente, Visto, Texto, Fecha\_msj)$

- $SUSCRITA(\overline{\underline{Id\_cuenta}}^{Cuenta}, \overline{\underline{Nombre\_tier}}^{Tier}, Fecha\_inicio, Fecha\_caducidad)$

  NOTA: Se incluye la fecha de inicio de suscripción al PK, ya que una cuenta puede suscribirse a un mismo tier varias veces (se puede suscribir otro tier si es que se caducó el anterior).

- $MANEJA(\overline{\underline{Nombre\_tier}}^{Tier}, \overline{\underline{Nombre\_permiso}}^{Permiso})$

- $LIKES(\overline{\underline{Id\_liker}}^{Cuenta}, \overline{\underline{Id\_liked}}^{Cuenta}, Super, Fecha\_like)$

  NOTA: Como ahora, el PK de perfil y el PK de preferencias es nada más id_cuenta, para mapear las relaciones binarias o ternarias y los atributos multivaluados asociados, solo requiere agregar el id_cuenta en lugar de agregarle también el id_perfil o id_pref.

- $SWIPES(\overline{\underline{Id\_disliker}}^{Cuenta}, \overline{\underline{Id\_disliked}}^{Cuenta})$

- $MATCH\_WITH(\overline{\underline{Id\_matcher}}^{Cuenta}, \overline{\underline{Id\_matched}}^{Cuenta}, Fecha)$

- $TRABAJA\_EN(\overline{\underline{Id\_cuenta}}^{Cuenta}, \overline{\underline{Id\_empresa}}^{Empresa}, Cargo, Fecha\_inicio)$

- $ESTUDIO\_EN(\overline{\underline{Id\_cuenta}}^{Cuenta}, \overline{\underline{Dominio}}^{Institucion}, Titulo, Anio\_ingreso, Anio\_egreso)$

- $CHATEA\_CON(\overline{\underline{Id\_chat}}^{Chat}, \overline{Id\_cuenta1}^{Cuenta}, \overline{Id\_cuenta2}^{Cuenta})$

- $REALIZA(\overline{Id\_cuenta}^{Cuenta}, \overline{\underline{Id\_pago}}^{Pago}, \overline{Digitos\_tarjeta}^{Tarjeta})$

- $ESTA\_EN\_AGRUPACION(\overline{\underline{Id\_cuenta}}^{Cuenta}, \overline{\underline{Dominio}}^{Institucion}, Agrupacion)$

- $TIENE\_HOBBY(\overline{\underline{Id\_cuenta}}^{Cuenta}, Hobby)$

- $TIENE\_HABILIDADES(\overline{\overline{Id\_cuenta, Habilidad}}^{Cuenta})$
- $TIENE\_FOTO(\overline{\overline{Id\_cuenta, Foto}}^{Cuenta})$
- $TIENE\_ORIENTACION\_SEXUAL(\overline{\overline{Id\_cuenta, Orientacion\_sexual}}^{Cuenta})$
- $TIENE\_CERTIFICACIONES(\overline{\overline{Id\_cuenta, Certificaciones}}^{Cuenta})$
- $ARCHIVO(\overline{\overline{Id\_chat, Numero\_msj, Nombre}}^{Mensaje}, Tipo, Contenido)$
- $PREF\_ORIENTACION\_SEXUAL(\overline{\overline{Id\_cuenta, Orientacion\_sexual}}^{Cuenta})$
- $PREF\_SEXO(\overline{\overline{Id\_cuenta, Sexo}}^{Cuenta})$

2. Definicion de DDL

Enlace directo al GitHub donde se encuentra el DDL del proyecto. Aqui.

3. Requerimientos Funcionales

1. Si un usuario se registra en RobleAffinity: Tinder para viejos egresados, se crea una cuenta en la tabla cuenta con

```
INSERT INTO cuenta (nombre, apellido, fecha_nacimiento, email,
                    contrasena, telefono)
VALUES ('Pedro', 'Pérez', '21/05/19YY', 'pedroperez2105@gmail.com',
        'hashcontraseña123', '5804XXXXXXXX');
```

   (Automáticamente se setea el id de la cuenta y la fecha de creación de la cuenta). Además, se le pide si desea configurar las notificaciones, el idioma y el tema de la app (el nuevo usuario selecciona el idioma Inglés, tema oscuro y notificaciones desactivadas).

```sql
UPDATE cuenta
SET idioma = 'eng', tema = FALSE, notificaciones = FALSE
WHERE email = 'pedroperez2105@gmail.com';
```

Además, se le pide al usuario su sexo y que active su ubicación en el celular para encontrar el latitud y la longitud que se encuentra este usuario. Y se le pide si desea llenar la descripción de su perfil o no (el usuario salta este paso). Entonces, se le crea un perfil para el usuario en la tabla perfil con

```sql
INSERT INTO perfil (id_cuenta, sexo, latitud, longitud)
VALUES ((SELECT id_cuenta FROM cuenta WHERE email =
'pedroperez2105@gmail.com'), 'M', -79.2349, 32.5894);
```

También, se le pide al usuario que suba al menos una foto en su perfil (se crea una instancia en tiene_foto asociado al perfil de este usuario) y se le pregunta si desea configurar sus preferencias. Si lo desea, se crea una instancia en la tabla de preferencias con los valores de preferencias proporcionadas. Si no desea configurarlo, entonces en su pantalla principal de la app del usuario, se le muestra cualquier tipo de egresado viejo que esté a lo sumo 5 km de él (por default).

2. Cuando se registra el usuario, también se le pide el nombre de la institución en que estudió y el título recibido, con su año de ingreso y egreso. Nota: La institución debe estar en la base de datos para que el usuario pueda seleccionarla. Si el usuario no encuentra su institución en la lista, puede seleccionar la opción 'OTRO' y proporcionar los datos de la misma. El sistema luego verificará si la institución existe y la registrará en la base de datos. Si se determina que la institución no existe, se notificará al usuario por correo y se eliminará su cuenta, explicando la razón de la eliminación.

3. En el app, el usuario solo puede modificar su email, contraseña y teléfono, idioma, notificaciones y tema. Y si modifica alguno de estos, se hace un UPDATE en cuenta. También, el usuario puede modificar su sexo y su descripción en el perfil. La ubicación también se puede modificar si es que tiene activado la ubicación de su celular. Y si el usuario pasó el proceso de verificación de su perfil, se hace un UPDATE de su perfil con verificado = TRUE. Solo se puede verificar el perfil exactamente una vez.

4. Si el usuario configura alguna de sus preferencias de los usuarios que quiere que le muestre en la pantalla principal de la app, se hace un UPDATE de los valores en la tabla de preferencias.

5. Si el usuario registra alguna tarjeta de crédito o débito en la app, se le crea una instancia en la tabla de tarjeta con los datos proporcionados. También puede eliminar

la tarjeta registrada si lo desea.

6. Si el usuario quiere suscribirse a un tier, se verifica que no esté suscrita a ninguna por los momentos. Si no tiene suscripciones activas, se procede a pedir al usuario que realice el pago con alguna de las tarjetas registradas. Luego del pago, se crea una instancia en la tabla de realiza y una instancia de los datos del pago realizado en pago.

   Cuando el pago realizado tenga estado = "Aprobado", se crea una instancia en la tabla de suscrita con una fecha de inicio del día en que fue aprobado el pago y una fecha de caducidad que es la suma entre la fecha de inicio + la duración en días del tier que se quiere suscribir.

7. Todo usuario con subscripción a un tier tiene la opción de aumentarla, para esto se verifica si el tier actual del usuario en la tabla suscrita posee capacidad de aumentar, es decir si el tier actual es 'Plus', se le da opción de 'Gold' y 'Platinum', en caso de ser 'Gold' entonces solo tendrá disponible 'Platinum'. Esto pasa con cualquier sistema de tiers, existirá una jerarquía implícita que dependerá de la cantidad de instancias en la tabla maneja asociadas al tier, mientras mas permisos tenga dicho tier, mayor rango tendrá en la jerarquía.

   Usuario que tenga el mayor tier con permisos vinculados no tendrá opción de actualización.

8. Todo usuario con subscripción a un tier 'Plus', o mayor, tiene la opción de actualizar las coordenadas origen en las preferencias. Esto se realizaría verificando que el usuario cuenta con al menos una instancia en la tabla suscrita que se encuentre vigente (Fecha_caducidad sea mayor a la fecha actual), luego el usuario tendrá la opción habilitada para realizar una actualización de la coordenada origen de sus preferencias que se realiza bajo un UPDATE sobre las referencias respectivas del usuario.

9. Si se quiere crear un nuevo tier, se proporciona el nombre del tier para agregarlo en la tabla de tier y todo tier debe tener al menos un permiso que maneje el nuevo tier en la tabla maneja.

10. Para agregar un nuevo permiso, se le proporciona el nombre y la descripción del tier y debe agregar al menos una instancia en maneja de este nuevo permiso con algún tier existente. Nota: Se crea primero el tier y luego los permisos de esta tier (si es que aún no están en la base de datos), posteriormente, se agrega las instancias de maneja entre el nuevo tier y los permisos.

11. El usuario también puede registrar en qué empresa está trabajando actualmente, con su cargo y fecha de inicio. Cuando el usuario registre la empresa, se crea una instancia en la tabla empresa y en trabaja_en.

12. Se hace un UPDATE si el usuario modifica la fecha de inicio o el cargo del trabajo en que está actualmente en la tabla trabaja_en.

13. Cuando el usuario dé like (swipe right) o dislike (swipe left) a otro usuario, se registra esto en la base de datos, en la tabla de likes y swipes, respectivamente. Para el like se le registra si fue o no un superlike y se registra también la fecha del like.

    Los usuarios que reciben un superlike reciben una notificación en la aplicación, para saber quienes están muy interesados en ellos.

14. Un usuario no puede eliminar el like ni el dislike que dió a otro usuario, al menos que esté suscrita a un tier.

15. Cuando dos usuarios se dan like el uno al otro, se crea una instancia en la tabla match de ellos dos, y se registra la fecha del match. Además, se crea un chat entre ellos y una instancia en chatea_con donde el id_cuenta1 es el usuario que dio primero el like al otro, en caso contrario sería el id_cuenta2.

16. Los mensajes de texto y los archivos que se mandan en los chats se registran en la base de datos. Los usuarios no pueden eliminar los mensajes en sus chats.

17. Cuando el remitente vea el mensaje, se hace un UPDATE del visto = True del mensaje.

18. El usuario también puede colocar las agrupaciones a las que perteneció durante sus estudios en la institución en el app (se crea una instancia en esta_en_agrupacion con los datos proporcionados). Así mismo, con los hobbies (en tiene_hobby), habilidades (en tiene_habilidades), fotos (en tiene_foto), certificaciones (en tiene_certificaciones), orientaciones sexuales (en tiene_orientacion_sexual), las preferencias en orientaciones sexuales (en pref_orientacion_sexual) y de género (en pref_sexo) de los usuarios que desea que le muestre en su pantalla principal.

    Si el usuario luego elimina alguna de estas, se elimina en la base de datos también, excepto con las fotos, el cual el usuario debe tener al menos una foto de él/ella en la base de datos (se le alerta al usuario que no puede eliminar la única foto que le queda).

19. Las suscripciones de los usuarios a los tiers permanecen en la base de datos incluso cuando se acabe la fecha de caducidad del usuario. Sin embargo, el usuario ya no tiene acceso a los permisos asociados con esa suscripción si la fecha actual es posterior a la fecha de caducidad.

20. Las instituciones no se pueden eliminar de la base de datos, ya que en el futuro puede haber usuarios que estudien en alguna de ellas. Solo se puede agregar y modificar datos de las instituciones.

21. Cuando el usuario realiza un pago dentro de la app, este se registra en la base de datos con los datos de su id (que se genera automáticamente), número de factura, estado (si fue aprobada o fallida), método de pago, monto, fecha y el documento de la factura.

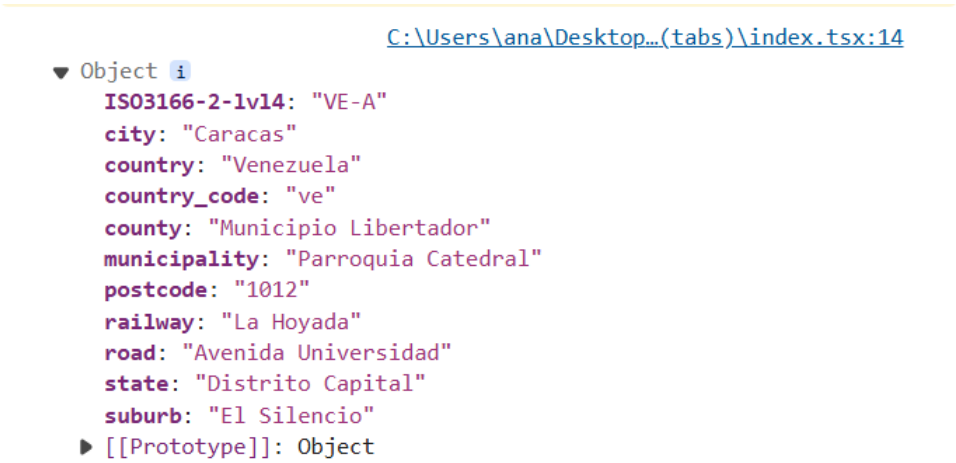
22. Puede crearse una institucion, para hacerlo deben agregarse los siguientes datos: dominio, nombre, tipo de institución, año de fundación, latitud y longitud.

23. Puede crearse una empresa, para hacerlo solo se necesita el nombre de la empresa, la clave será un identificador autogenerado.

24. En el perfil del usuario, se le muestra sus fotos, nombre, apellido, edad (se hace un cálculo de la diferencia en años de la fecha actual con la fecha de nacimiento), sexo, la ciudad y país en que se encuentra, si el perfil está verificada o no, hobbies, descripción, certificaciones, instituciones en que estudió, habilidades, empresas en que trabaja actualmente y, orientaciones sexuales.

25. Para mostrar la ciudad y país de un usuario(perfil), institución o en preferencias, se hace uso de la herramienta Nominatim con la latitud y longitud.

```
const [add, setAdd] = useState<{ city: string, country: string }>({ city: '', country: '' });
const latitude = 10.503316;
const longitud = -66.91117567613144;
const url = `https://nominatim.openstreetmap.org/reverse?format=json&lat=${latitude}&lon=${longitude}`;
useEffect(()=>{
  fetch(url).then(res=>res.json()).then(data=>setAdd(data.address))
}, [])
console.log(add)
```



26. La edad mostrada en un perfil se determina por la fecha de nacimiento que el usuario proporciono al momento de crearse una cuenta en el sistema.

27. Los numero_likes de una perfil es determinado por contar cuántas veces aparece el id_cuenta en la columna id_liked de la tabla likes.

28. Los likes_por_día de un perfil es determinado por contar cuántas veces aparece el id_cuenta en la columna id_liker en asociación con la fecha acorde al día respectivo en la tabla likes.

29. Para determinar los usuarios que estén a una distancia máxima de un determinado usuario se hace uso de una función de postgis: ST_DistanceSphere()

    Ejemplo de cómo determinar las instituciones que estén a una distancia máxima de 200km de una institución con dominio = 'example.com'

```
SELECT *
FROM institucion
WHERE ST_DistanceSphere(
      coordenada,
      (SELECT coordenada FROM institucion WHERE dominio =
'example1.com')
      ) / 1000 <= 200;
```

30. TOMA DE DECISIÓN 1: Cuando una cuenta es eliminada del sistema se procede a mantener los registros de pagos que esa cuenta a realizado en todo su periodo de existencia en la aplicación. Esta decisión tiene el propósito de tener un respaldo legal de cualquier insurgencia futura para con la empresa por parte de la persona a cargo de la cuenta eliminada.

31. TOMA DE DECISIÓN 2: Cuando una cuenta es eliminada, los chats donde dicha cuenta participa se verá reflejado como ”Cuenta eliminada”, es decir la otra cuenta existente verá que los mensajes vienen de una ”Cuenta eliminada”, permitiendo así tener registro de la conversación que la cuenta existente tuvo con la cuenta eliminada. Esta decisión tiene el propósito de tener un respaldo legal de cualquier abuso, acoso, fraude, delitos o cualquier otra irrupción de índole social, económica o legal.