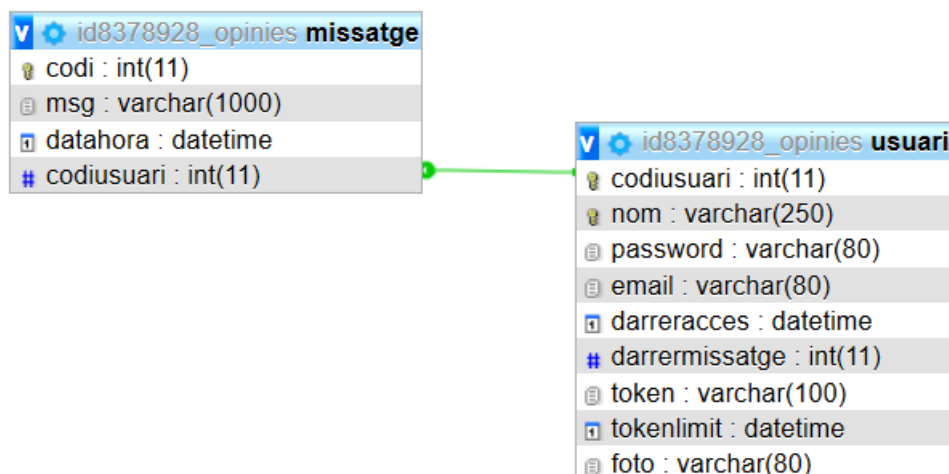




Projecte missatgeria. Base de dades.

La base de dades del projecte inicialment es molt senzilla, està formada per dues taules, una taula amb els missatges i la taula de usuaris.



darrermissatge: guarda quin es el darrer missatge que l'usuari ha descarregat. Així quan feim una petició al servidor tan sols descarregam els missatges no llegits.

<https://iesmantpc.000webhostapp.com/public/missatge/{idusuari}>

token,tokenlimit: ho farem servir per mes endavant validar l'accés al recurs.

Crearem una base de dades SQLite per guardar els missatges descarregats (i també els usuaris) de manera local.

Després del login (a partir de qual obtindrem el codi del nostre usuari necessari per descarregar els nostres missatges) el primer que hauríem de fer és un mètode que

1. Descarregui els missatges.
2. Els guardi a la nostra base de dades.
3. Visualitzi els missatges que hi ha a la base de dades amb un listview.

Per fer proves podem fer servir:

<https://iesmantpc.000webhostapp.com/public/provamissatge/>

Això descarrega tots els missatges i no necessita que enviem el token de seguretat. La petició esmentada anteriorment sols descarrega missatges si n'hi ha de nous.

Una possible implementació del SQLiteOpenHelper per la creació de l base de dades local pot esser:

```
public class HelperQuepassaeh extends SQLiteOpenHelper {
    public static final String TABLE_MISSATGE = "missatge";
    public static final String COLUMN_CODI = "codi";
    public static final String COLUMN_MSG = "msg";
    public static final String COLUMN_DATAHORA = "datahora";
    public static final String COLUMN_FKCODIUSUARI = "codiusuari";
    public static final String COLUMN_PENDENT = "pendent";
    public static final String TABLE_USUARI = "usuari";
    public static final String COLUMN_CODIUSUARI = "codiusuari";
    public static final String COLUMN_NOM = "nom";
    public static final String COLUMN_EMAIL = "email";
    public static final String COLUMN_FOTO = "foto";
    private static final String DATABASE_NAME = "quepassaeh.db";
    private static final int DATABASE_VERSION = 6;
    private static final String DATABASE_CREATE_MISSATGE = "create table "
        + TABLE_MISSATGE + "(" + COLUMN_CODI
        + " integer primary key, "
        + COLUMN_MSG + " text not null,"
        + COLUMN_DATAHORA + " text not null,"
        + COLUMN_FKCODIUSUARI + " integer not null,"
        + COLUMN_PENDENT + " integer default 0 not null,"
        + "FOREIGN KEY(" + COLUMN_FKCODIUSUARI + ") REFERENCES
"+TABLE_USUARI+"(" + COLUMN_CODIUSUARI + "))";
    private static final String DATABASE_CREATE_USUARI = "create table "
        + TABLE_USUARI + "(" + COLUMN_CODIUSUARI
        + " integer primary key, "
        + COLUMN_NOM + " text not null,"
        + COLUMN_EMAIL + " text," + COLUMN_FOTO + " text)";
    public HelperQuepassaeh(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase database) {
        database.execSQL(DATABASE_CREATE_USUARI);
        Log.i("QuePassaEh", DATABASE_CREATE_MISSATGE);
        database.execSQL(DATABASE_CREATE_MISSATGE);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        Log.w(HelperQuepassaeh.class.getName(),
            "Upgrading database from version " + oldVersion + " to "
            + newVersion + ", which will destroy all old data");
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_MISSATGE);
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_USUARI);
        onCreate(db);
    }
}
```