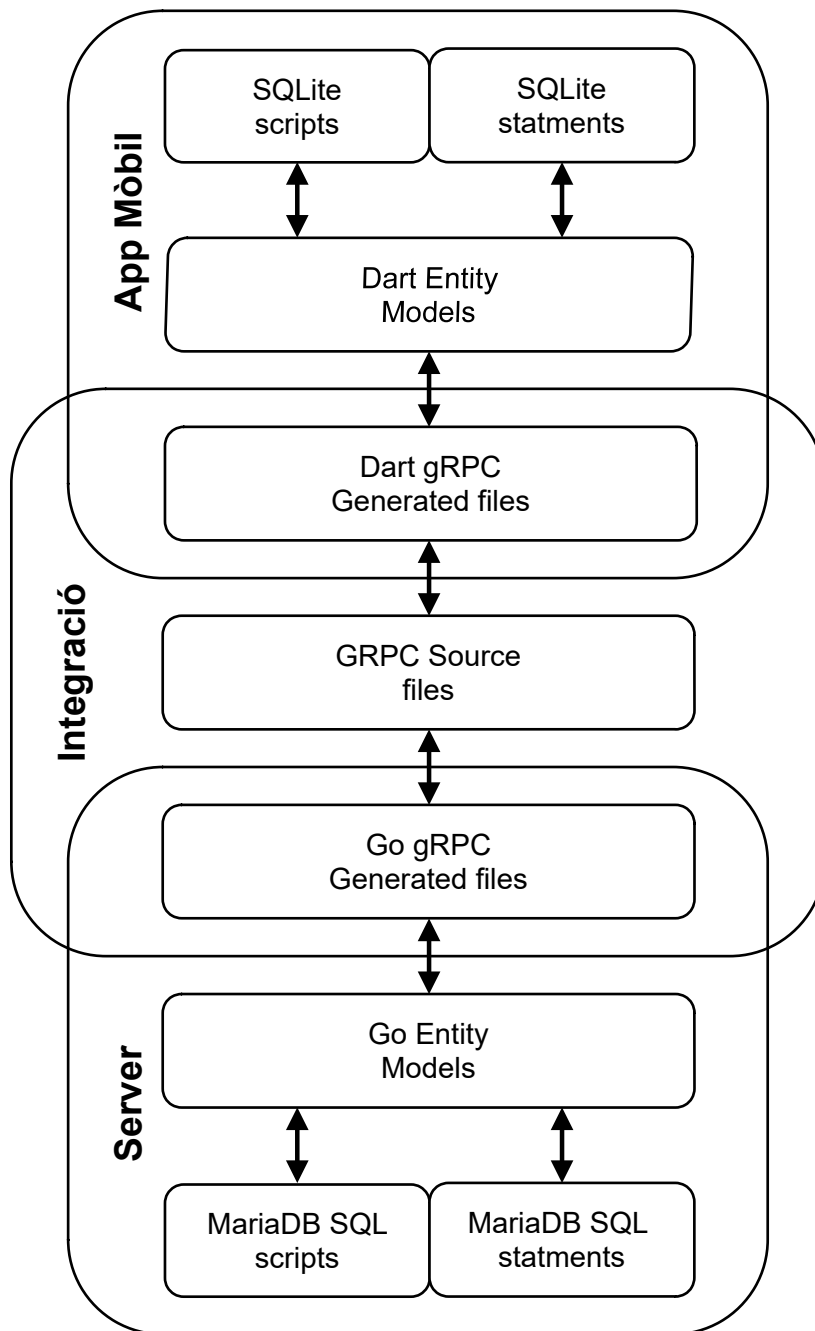


# Plataforma Sabina

## Model de Dades - Implementació

versió 0.7.2

2024-09-07 ds



## Contingut

Implementació del Model de Dades.....	1
Servidor.....	1
APP mòbil.....	1
Integració.....	1
Diagrama lògic.....	2
1 Estructures Auxiliars.....	3
2 Mòdul USRMOD.....	4
2.1 Entitat USER.....	4
2.1.1 Proto3.....	4
2.1.2 Flutter.....	4
2.1.3 Golang.....	4
2.1.4 SQLite.....	5
2.1.5 MariaDB.....	5
2.2 Entitat DEVICE.....	6
2.2.1 Proto3.....	6
2.2.2 Flutter.....	6
2.2.3 Golang.....	6
2.2.4 SQLite.....	7
2.2.5 MariaDB.....	7
2.3 Entitat FCM_HISTORY.....	7
2.3.1 Proto3.....	7
2.3.2 Flutter.....	7
2.3.3 Golang.....	7
2.3.4 SQLite.....	8
2.3.5 MariaDB.....	8
3 Mòdul DISMOD.....	9
3.1 Entitat DSM_V.....	9
3.1.1 Proto3.....	9
3.1.2 Flutter.....	9
3.1.3 Golang PENDENT.....	9
3.1.4 SQLite.....	9
3.1.5 MariaDB.....	9
3.2 Entitat DISEASE.....	9
3.2.1 Proto3.....	9
3.2.2 Flutter.....	10
3.2.3 Golang.....	10
3.2.4 SQLite.....	10
3.2.5 MariaDB.....	10
3.3 Entitat PHASE.....	10
3.3.1 Proto3.....	10
3.3.2 Flutter.....	10
3.3.3 Golang.....	11
3.3.4 SQLite.....	11
3.3.5 MariaDB.....	11
3.4 Entitat GOAL.....	11
3.4.1 Proto3.....	11

3.4.2 Flutter.....	11
3.4.3 Golang.....	11
3.4.4 SQLite.....	12
3.4.5 MariaDB.....	12
4 Mòdul EMOMOD.....	13
4.1 Entitat EMOTION.....	13
4.1.1 Proto3.....	13
4.1.2 Flutter.....	13
4.1.3 Golang.....	13
4.1.4 SQLite.....	13
4.1.5 MariaDB.....	13
4.2 Entitat MOOD.....	14
4.2.1 Proto3.....	14
4.2.2 Flutter.....	14
4.2.3 Golang.....	14
4.2.4 SQLite.....	14
4.2.5 MariaDB.....	14
5 Mòdul RSCMOD.....	15
5.1 Entitat RESOURCE.....	15
5.1.1 Proto3.....	15
5.1.2 Flutter.....	15
5.1.3 Golang.....	15
5.1.4 SQLite.....	15
5.1.5 MariaDB.....	16
5.2 Entitat PHASE_RESOURCE.....	16
5.2.1 Proto3.....	16
5.2.2 Flutter.....	16
5.2.3 Golang.....	16
5.2.4 SQLite.....	16
5.2.5 MariaDB.....	16
6 Mòdul TCKMOD.....	17
6.1 Entitat TRACKING.....	17
6.1.1 Proto3.....	17
6.1.2 Flutter.....	17
6.1.3 Golang.....	17
6.1.4 SQLite.....	17
6.1.5 MariaDB.....	17
6.2 Entitat PHASE_TRACKING.....	18
6.2.1 Proto3.....	18
6.2.2 Flutter.....	18
6.2.3 Golang.....	18
6.2.4 SQLite.....	18
6.2.5 MariaDB.....	18
6.3 Entitat TRACKING_COLUMN.....	18
6.3.1 Proto3.....	18
6.3.2 Flutter.....	19
6.3.3 Golang.....	19
6.3.4 SQLite.....	19

6.3.5 MariaDB.....	19
7 Mòdul TSTMOD.....	20
7.1 Entitat TEST_CATEGORY.....	20
7.1.1 Proto3.....	20
7.1.2 Flutter.....	20
7.1.3 Golang.....	20
7.1.4 SQLite.....	20
7.1.5 MariaDB.....	20
7.2 Entitat TEST.....	20
7.2.1 Proto3.....	20
7.2.2 Flutter.....	21
7.2.3 Golang.....	21
7.2.4 SQLite.....	21
7.2.5 MariaDB.....	21
7.3 Entitat TEST_QUESTION.....	22
7.3.1 Proto3.....	22
7.3.2 Flutter.....	22
7.3.3 Golang.....	22
7.3.4 SQLite.....	22
7.3.5 MariaDB.....	23
8 Mòdul DGNMOD.....	24
8.1 Entitat DIAGNOSIS.....	24
8.1.1 Proto3.....	24
8.1.2 Flutter.....	24
8.1.3 Golang.....	24
8.1.4 SQLite.....	24
8.1.5 MariaDB.....	24
8.2 Entitat DIAGNOSIS_PHASE.....	25
8.2.1 Proto3.....	25
8.2.2 Flutter.....	25
8.2.3 Golang.....	25
8.2.4 SQLite.....	25
8.2.5 MariaDB.....	25
8.3 Entitat ACHIEVEMENT.....	26
8.3.1 Proto3.....	26
8.3.2 Flutter.....	26
8.3.3 Golang.....	26
8.3.4 SQLite.....	26
8.3.5 MariaDB.....	27
9 Mòdul MATMOD.....	28
9.1 Entitat MATERIAL.....	28
9.1.1 Proto3.....	28
9.1.2 Flutter.....	28
9.1.3 Golang.....	28
9.1.4 SQLite.....	28
9.1.5 MariaDB.....	28
9.2 Entitat MATERIAL_PHASE.....	29
9.2.1 Proto3.....	29

9.2.2 Flutter.....	29
9.2.3 Golang.....	29
9.2.4 SQLite.....	29
9.2.5 MariaDB.....	29
10 Mòdul REGMOD.....	30
10.1 Entitat REGISTER.....	30
10.1.1 Proto3.....	30
10.1.2 Flutter.....	30
10.1.3 Golang.....	30
10.1.4 SQLite.....	30
10.1.5 MariaDB.....	30
10.2 Entitat REGISTER_COLUMN.....	31
10.2.1 Proto3.....	31
10.2.2 Flutter.....	31
10.2.3 Golang.....	31
10.2.4 SQLite.....	31
10.2.5 MariaDB.....	31
11 Mòdul RESMOD.....	32
11.1 Entitat PATIENT_TEST.....	32
11.1.1 Proto3.....	32
11.1.2 Flutter.....	32
11.1.3 Golang.....	32
11.1.4 SQLite.....	32
11.1.5 MariaDB.....	32
11.2 Entitat ANSWER.....	33
11.2.1 Proto3.....	33
11.2.2 Flutter.....	33
11.2.3 Golang.....	33
11.2.4 SQLite.....	33
11.2.5 MariaDB.....	34
12 Mòdul NTFMOD.....	35
12.1 Entitat NOTIFICATION.....	35
12.1.1 Proto3.....	35
12.1.2 Flutter.....	35
12.1.3 Golang.....	35
12.1.4 SQLite.....	35
12.1.5 MariaDB.....	36
13 Mòdul TSKMOD.....	37
13.1 Entitat TASK.....	37
13.1.1 Proto3.....	37
13.1.2 Flutter.....	37
13.1.3 Golang.....	37
13.1.4 SQLite.....	37
13.1.5 MariaDB.....	38
14 Mòdul VISMODO.....	39
14.1 Entitat VISIT.....	39
14.1.1 Proto3.....	39
14.1.2 Flutter.....	39

14.1.3 Golang.....	39
14.1.4 SQLite.....	39
14.1.5 MariaDB.....	39
15 Mòdul LOCMOD.....	41
15.1 Entitat TRANSLATION.....	41
15.1.1 Proto3.....	41
15.1.2 Flutter.....	41
15.1.3 Golang.....	41
15.1.4 SQLite.....	41
15.1.5 MariaDB.....	41
16 Mòdul LSTMOD.....	42
16.1 Entitat LIST_CATEGORY.....	42
16.1.1 Proto3.....	42
16.1.2 Flutter.....	42
16.1.3 Golang.....	42
16.1.4 SQLite.....	42
16.1.5 MariaDB.....	42
16.2 Entitat OPTION_LIST.....	42
16.2.1 Proto3.....	42
16.2.2 Flutter.....	43
16.2.3 Golang.....	43
16.2.4 SQLite.....	43
16.2.5 MariaDB.....	43
16.3 Entitat OPTION_ENTRY.....	44
16.3.1 Proto3.....	44
16.3.2 Flutter.....	44
16.3.3 Golang.....	44
16.3.4 SQLite.....	44
16.3.5 MariaDB.....	44
17 Índexs.....	45

## Implementació del Model de Dades

El propòsit d'aquest document és documentar com les diferents entitats de dades son implementats dins els diferents subsistemes que existeixen al projecte.

Els subsistemes del projecte són:

### Servidor

El subsistema **Servidor** és el nucli de la plataforma on s'enregistren i gestionen totes les dades consolidades.

Aquest subsistema s'implementa en un host *Linux* (família i versió per acabar de determinar segons les necessitats), en llenguatge *Golang*, fent servir un DBSM *MariaDB v11.5.2 Rolling*.

### APP mòbil

El subsistema **App** és la interfície d'accés a la plataforma de tots els tipus d'usuari.

Aquest subsistema s'implementa en *Flutter* (versió per determinar) per a totes les plataformes mòbils (Android i iOS), fent servir un DBSM local SQLite (a través del paquet Dart *sqflite\_common: ^2.5.4+2*).

### Integració

El subsistema d'**Integració** s'encarrega de la comunicació de dades segura entre els dispositius mòbils i el servidor.

Per a generalitzar el model de base de dades dins aquest subsistema s'utilitza *gRPC* (Google Remote Procedure Call).

Tot i que la plataforma Sabina es recolza sobre la plataforma *Firebase* de Google, no s'emmagatzema cap dada al seu sistema d'emmagatzemament en el núvol.

Diagrama lògic

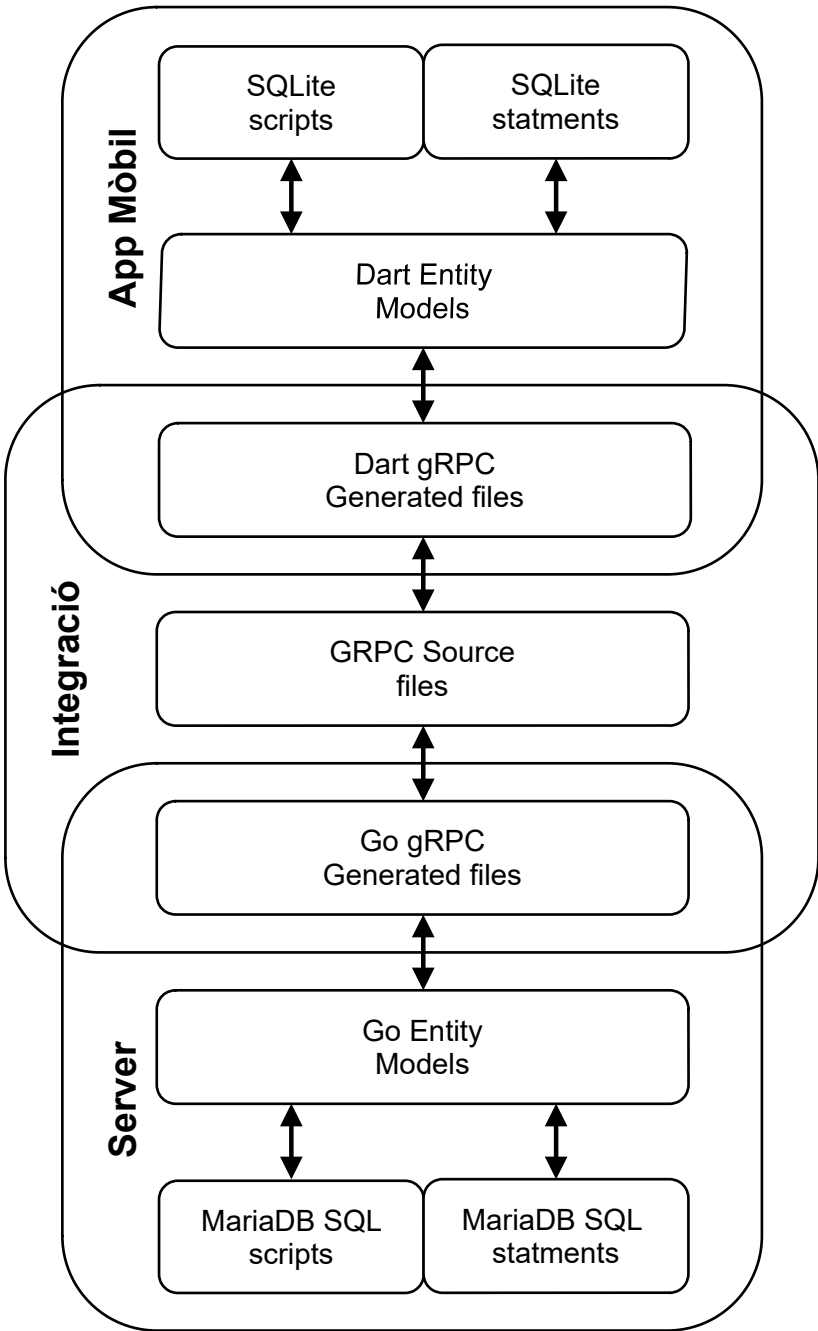


Diagrama 1: Implementació del Model de Dades



# 1 Estructures Auxiliars

Les properes definicions proveeixen les diferents vistes de cada entitat del model de dades segons els subsistemes.

En les definicions la majoria de les entitats faran servir una subentitat anomenada *BaseEntity* (en Integració), *ModelEntity* (en Flutter) i *GoBaseEntity* (en Go). Aquesta subentitat és un mecanisme de compactació de la descripció del model útil perquè la gran majoria de les entitats comparteixen un conjunt de propietats que es poden abstraure:

```
// Proto3
message BaseEntity {
  optional string    local_key = 1;
  optional string    server_key = 2;
  uint32             created_by = 3;
  Timestamp           created_at = 4;
  optional uint32     updated_by = 5;
  optional Timestamp  updated_at = 6;
  bool               is_new = 7;
  bool               is_updated = 8;
  bool               is_deleted = 9;
}
```

```
// Flutter
abstract class ModelEntity {
  late CoreEntity _core;
}

class CoreEntity {
  late int? _localId;
  late int? _id;
  late User? _createdBy;
  late DateTime? _createdAt;
  late User? _updatedBy;
  late DateTime? _updatedAt;
  late bool _isNew;
  late bool _isUpdated;
  late bool _isDeleted;
}
```

```
// Golang
type GoBaseEntity struct {
  GoEntity
  LocalId      *uint32
  ServerId     *uint32
  CreatedBy    uint32
  CreatedAt    dt.Time
  UpdatedBy    *uint32
  UpdatedAt    *dt.Time
  InstIsNew    bool
  InstIsUpdated bool
  InstIsDeleted bool
}
```

## 2 Mòdul USRMOD

### 2.1 Entitat USER

#### 2.1.1 Proto3

```
message UsrUser {
  BaseEntity          base_entity = 1;
  UsrUserType         user_type = 2;
  UsrUserState        state = 3;
  uint64              permissions = 4;
  string              alias = 5;
  optional bytes      certificate = 6;
  optional Timestamp  birth_date = 7;
  optional uint64     first_key = 8;
  optional Timestamp  first_conn_at = 9;
  string              locale = 10;
  optional uint32     therapist_id = 11;
}
```

#### 2.1.2 Flutter

```
enum UserType {
  unspecified,
  root,
  admin,
  patient,
  therapist,
}
```

```
enum UserState {
  unspecified,
  building,
  active,
  locked,
}
```

```
class UsrUser extends ModelEntity {
  UserType _usrType = UserType.unspecified;
  UserState _usrState = UserState.unspecified;
  Uint64 _permissions = 0 as Uint64;
  String? _alias;
  Uint8List? _certificate;
  DateTime? _birthDate;
  Uint64? _firstConnKey;
  DateTime? _firstConnAt;
  UsrDevice? _device;
  UsrUser? _therapist;
  Locale _locale = localeES;
}
```

#### 2.1.3 Golang

```
type UserType int
const (
  User_type_unspecified UserType = 0
  User_type_root         UserType = 1
  User_type_admin        UserType = 2
  User_type_patient      UserType = 3
  User_type_therapist     UserType = 4
)
```

```
type UserState int
const (
  User_state_unspecified UserState = 0
  User_state_building    UserState = 1
  User_state_active      UserState = 2
  User_state_locked      UserState = 3
)
```

```

type UsrUser struct {
    Entity      GoBaseEntity
    GoProto
    GoSQL
    UType       UserType
    State       UserState
    Permissions  uint64
    Alias       string
    Certificate  *x509.Certificate
    BirthDate   *dt.Time
    FirstConnPKey *uint64
    FirstConnAt *dt.Time
    Device      *UsrDevice
    TherapistId *uint32
    Locale      LocLocale
}

```

### 2.1.4 SQLite

```

CREATE TABLE $tnUsrUser (
    $standardHeader,
    $fldUserType      $dbtIntNotNull,
    $fldUserState     $dbtIntNotNull,
    $fldPermissions   $dbtIntNotNull DEFAULT 0,
    $fldAlias         $dbtTextNotNull,
    $fldCertificate   $dbtText,
    $fldBirthDate     $dbtDateTime,
    $fldFirstConnKey  $dbtInt,
    $fldFirstConnAt   $dbtDateTime,
    $fldLocaleCode    $dbtTextNotNull DEFAULT "es",
    $fldTherapist     $dbtInt REFERENCES $tnUsrUser($fldId)
);
ALTER TABLE $tnUsrUser ADD COLUMN $fldDevice $dbtInt REFERENCES $tnUsrDevice($fldId);

```

### 2.1.5 MariaDB

```

CREATE TABLE USER (
    ID                INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY        INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT        DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY        INT UNSIGNED NOT NULL REFERENCES USER(ID),
    UPDATED_AT        DATETIME,
    USER_TYPE          INT UNSIGNED NOT NULL,
    USER_STATE         INT UNSIGNED NOT NULL,
    PERMISSIONS        INT UNSIGNED NOT NULL DEFAULT 0,
    ALIAS              VARCHAR(100) NOT NULL,
    CERTIFICATE        BLOB,
    BIRTH_DATE         DATE,
    FIRST_KEY          INT,
    FIRST_CONN_AT      DATETIME,
    LOCALE_CODE        VARCHAR(2) NOT NULL DEFAULT @loc_esid,
    THERAPIST_ID       INT UNSIGNED REFERENCES USER(ID),
    INDEX (CREATED_BY)
) ENGINE = InnoDB;
ALTER TABLE USER1 ADD COLUMN DEVICE_ID INT UNSIGNED REFERENCES DEVICE(ID);

```

<sup>1</sup> Donat que les instàncies d'USER i DEVICE estableixen una referència creuada, primer es crea la taula USER, després es crea la taula DEVICE i després s'afegeix la referència al dispositiu a la taula d'usuari.

## 2.2 Entitat DEVICE

### 2.2.1 Proto3

```
message UsrDevice {
  BaseEntity      base_entity = 1;
  UsrDeviceType   type = 2;
  UsrDeviceState  state = 3;
  optional string token = 4;
  uint32          owner = 5;
}
```

### 2.2.2 Flutter

```
enum DeviceType {
  unspecified,
  androidMobile,
  androidTablet,
  iosMobile,
  iosTablet,
  windows,
  macOS,
  linux,
}
```

```
enum DeviceState {
  unspecified,
  enabled,
  disabled,
}
```

```
class UsrDevice extends ModelEntity {
  DeviceType _devType = DeviceType.unspecified;
  DeviceState _devState = DeviceState.unspecified;
  String? _token;
  UsrUser? _owner;
}
```

### 2.2.3 Golang

```
type DeviceType int
const (
  device_type_unspecified DeviceType = 0
  device_type_mobileAndroid DeviceType = 100
  device_type_mobileIOS DeviceType = 200
  device_type_tabletAndroid DeviceType = 300
  device_type_tabletIOS DeviceType = 400
  device_type_windows DeviceType = 500
  device_type_macos DeviceType = 600
  device_type_linux DeviceType = 700
)
```

```
type DeviceState int
const (
  device_state_unspecified DeviceState = 0
  device_state_enabled DeviceState = 100
  device_state_disabled DeviceState = 200
)
```

```
type UsrDevice struct {
  Entity      GoBaseEntity
  GoProto
  GoSQL
  DType      DeviceType
  State      DeviceState
  FcmToken   *string
  Owner      uint32
}
```

### 2.2.4 SQLite

```
CREATE TABLE $tnUsrDevice (
    $standardHeader,
    $fldDeviceType $dbtIntNotNull,
    $fldDeviceState $dbtIntNotNull,
    $fldToken       $dbtTextNotNullUnique,
    $fldOwner       $dbtInt REFERENCES $tnUsrUser($fldId)
);
```

### 2.2.5 MariaDB

```
CREATE TABLE DEVICE (
    ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),
    UPDATED_AT  DATETIME,
    DEVICE_TYPE INT UNSIGNED NOT NULL,
    DEVICE_STATE INT UNSIGNED NOT NULL,
    FCM_TOKEN    VARCHAR(2048),
    OWNER_ID     INT UNSIGNED NOT NULL REFERENCES USER(ID)
) ENGINE = InnoDB;
```

## 2.3 Entitat FCM\_HISTORY

### 2.3.1 Proto3

```
message UsrFcmHistory {
    BaseEntity base_entity = 1;
    string      token = 2;
    UsrDevice   device = 3;
    optional    Timestamp since = 4;
    bool        last_one = 5;
}
```

### 2.3.2 Flutter

```
class UsrFcmHistory extends ModelEntity {
    String? _fcmToken;
    UsrDevice? _device;
    DateTime? _since;
    bool      _lastOne = false;
}
```

### 2.3.3 Golang

```
type UsrFcmHistory struct {
    Entity    GoBaseEntity
    GoProto
    GoSQL
    FcmToken  *string
    Device    UsrDevice
    Since     *dt.Time
    IsCurrent bool
    Owner     UsrUser
}
```

### 2.3.4 SQLite

```
CREATE TABLE $tnUsrFcmHistory (  
    $standardHeader,  
    $fldToken    $dbtTextNotNull UNIQUE,  
    $fldSince    $dbtDateTimeNotNull,  
    $fldLastOne  $dbtIntNotNull DEFAULT 0,  
    $fldDevice   $dbtInt REFERENCES $tnUsrDevice($fldId)  
);
```

### 2.3.5 MariaDB

```
CREATE TABLE FCM_HISTORY (  
    ID            INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    CREATED_BY   INT UNSIGNED NOT NULL REFERENCES USER(ID),  
    CREATED_AT   DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    UPDATED_BY   INT UNSIGNED REFERENCES USER(ID),  
    UPDATED_AT   DATETIME,  
    FCM_TOKEN    VARCHAR(2048) NOT NULL,  
    DEVICE_ID    INT UNSIGNED NOT NULL REFERENCES DEVICE(ID),  
    SINCE        DATETIME,  
    LAST_ONE     BOOLEAN NOT NULL DEFAULT FALSE  
) ENGINE = InnoDB;
```

## 3 Mòdul DISMOD

### 3.1 Entitat DSM\_V

Aquesta entitat es diferencia de la resta en que és un magatzem estàtic de dades. Per aquesta raó no disposa dels camps de registre sobre creació i actualització.

#### 3.1.1 Proto3

```
message DisDsmV {
  int32 id = 1;
  string name = 2;
  string icd10 = 3;
}
```

#### 3.1.2 Flutter

```
class DisDsmV extends ModelEntity {
  late String? _name;
  late String? _icd10;
}
```

#### 3.1.3 Golang PENDENT

#### 3.1.4 SQLite

```
CREATE TABLE $tnDisDsmV (
  $fldIdLocal $dbtIntID,
  $fldId      $dbtIntNotNullUnique,
  $fldName    $dbtTextNotNullUnique,
  $fldIcd10   $dbtTextNotNull
);
```

#### 3.1.5 MariaDB

```
CREATE TABLE DSM_V (
  ID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  DNAME VARCHAR(400) NOT NULL,
  CIE_10 VARCHAR(20) NOT NULL,
  FULLTEXT (DNAME),
  FULLTEXT (CIE_10)
) ENGINE = InnoDB;
```

## 3.2 Entitat DISEASE

### 3.2.1 Proto3

```
message DisDisease {
  BaseEntity    base_entity = 1;
  string        name_key = 2;
  optional string desc_key = 3;
  DisDsmV       dsmv = 4;
  UsrUser       theraphist = 5;
}
```

### 3.2.2 Flutter

```
class DisDisease extends ModelEntity {
  String? _nameKey;
  String? _name;
  String? _descKey;
  String? _desc;
  DisDsmV? _dsmV;
  UsrUser? _therapist;
}
```

### 3.2.3 Golang

PENDENT

### 3.2.4 SQLite

```
CREATE TABLE $tnDisDisease (
  $standardHeader,
  $fldNameKey $dbtTextNotNullUnique,
  $fldDescKey $dbtTextNotNull,
  $fldDsmV $dbtIntNotNull REFERENCES $tnDisDsmV($fldId),
  $fldTherapist $dbtIntNotNull REFERENCES $tnUsrUser($fldId)
);
```

### 3.2.5 MariaDB

```
CREATE TABLE DISEASE (
  ID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT DATETIME,
  NAME_KEY VARCHAR(100) NOT NULL,
  DESCRIPTION_KEY VARCHAR(100),
  DSM_V INT UNSIGNED NOT NULL REFERENCES DSM_V(ID),
  THERAPIST INT UNSIGNED NOT NULL REFERENCES USER(ID),
  INDEX (DSM_V)
) ENGINE = InnoDB;
```

## 3.3 Entitat PHASE

### 3.3.1 Proto3

```
message DisPhase {
  BaseEntity base_entity = 1;
  int32 idx = 2;
  string name_key = 3;
  optional string desc_key = 5;
  DisDisease disease = 4;
}
```

### 3.3.2 Flutter

```
class DisPhase extends ModelEntity {
  int? _idx;
  String? _nameKey;
  String? _name;
  String? _descKey;
  String? _desc;
  DisDisease? _disease;
}
```



```
}
```

### 3.3.3 Golang

PENDENT

### 3.3.4 SQLite

```
CREATE TABLE $tnDisPhase (
    $standardHeader,
    $fldIdx      $dbtIntNotNull,
    $fldNameKey  $dbtTextNotNull,
    $fldDescKey  $dbtTextNotNull,
    $fldDisease  $dbtIntNotNull REFERENCES $tnDisDisease($fldId)
);
```

### 3.3.5 MariaDB

```
CREATE TABLE PHASE (
    ID                INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY        INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT        DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY        INT UNSIGNED REFERENCES USER(ID),
    UPDATED_AT        DATETIME,
    IDX               INT UNSIGNED NOT NULL,
    NAME_KEY          VARCHAR(100) NOT NULL,
    DESCRIPTION_KEY    VARCHAR(100),
    DISEASE_ID        INT UNSIGNED NOT NULL REFERENCES DISEASE(ID),
    INDEX (DISEASE_ID, IDX)
) ENGINE = InnoDB;
```

## 3.4 Entitat GOAL

### 3.4.1 Proto3

```
message DisGoal {
    BaseEntity      base_entity = 1;
    int32           idx = 2;
    string          name_key = 3;
    optional string desc_key = 5;
    DisPhase        phase = 4;
}
```

### 3.4.2 Flutter

```
class DisGoal extends ModelEntity {
    int?      _idx;
    String?   _nameKey;
    String?   _name;
    String?   _descKey;
    String?   _desc;
    DisPhase? _phase;
}
```

### 3.4.3 Golang

PENDENT

#### 3.4.4 SQLite

```
CREATE TABLE $tnDisGoal (  
    $standardHeader,  
    $fldIdx          $dbtIntNotNull,  
    $fldNameKey      $dbtTextNotNull,  
    $fldDescKey      $dbtTextNotNull,  
    $fldDiseasePhase $dbtIntNotNull REFERENCES $tnDisPhase($fldId)  
);
```

#### 3.4.5 MariaDB

```
CREATE TABLE GOAL (  
    ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),  
    CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),  
    UPDATED_AT  DATETIME,  
    IDX         INT UNSIGNED NOT NULL,  
    NAME_KEY    VARCHAR(100) NOT NULL,  
    DESC_KEY    VARCHAR(100),  
    PHASE_ID    INT UNSIGNED NOT NULL REFERENCES PHASE(ID),  
    INDEX (PHASE_ID, IDX)  
) ENGINE = InnoDB;
```

## 4 Mòdul EMOMOD

### 4.1 Entitat EMOTION

#### 4.1.1 Proto3

```
message EmoEmotion {
    BaseEntity          base_entity = 1;
    string               name_key = 2;
    optional string      desc_key = 3;
    optional EmoEmotion parent = 4;
    int32                value = 5;
}
```

#### 4.1.2 Flutter

```
class EmoEmotion extends ModelEntity {
    String?      _nameKey;
    String?      _name;
    String?      _descKey;
    String?      _desc;
    EmoEmotion? _parent;
    int?         _value;
}
```

#### 4.1.3 Golang PENDENT

#### 4.1.4 SQLite

```
CREATE TABLE $tnEmoEmotion (
    $standardHeader,
    $fldNameKey $dbtTextNotNull,
    $fldDescKey $dbtText,
    $fldParent  $dbtInt REFERENCES $tnEmoEmotion($fldId),
    $fldValue   $dbtIntNotNull DEFAULT 0
);
```

#### 4.1.5 MariaDB

```
CREATE TABLE EMOTION (
    ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),
    UPDATED_AT  DATETIME,
    NAME_KEY    VARCHAR(100) NOT NULL,
    DESC_KEY    VARCHAR(100),
    PARENT_ID   INT UNSIGNED REFERENCES EMOTION(ID),
    VALUE       INT UNSIGNED NOT NULL
) ENGINE = InnoDB;
```

## 4.2 Entitat MOOD

### 4.2.1 Proto3

```
message EmoMood {
    BaseEntity      base_entity = 1;
    string           name_key = 2;
    optional string desc_key = 3;
    int32            value = 4;
}
```

### 4.2.2 Flutter

```
class EmoMood extends ModelEntity {
    String? _nameKey;
    String? _name;
    String? _descKey;
    String? _desc;
    int?    _value;
}
```

### 4.2.3 Golang

PENDENT

### 4.2.4 SQLite

```
CREATE TABLE $tnEmoMood (
    $standardHeader,
    $fldNameKey $dbtTextNotNull,
    $fldDescKey $dbtTextNotNull,
    $fldValue   $dbtIntNotNull DEFAULT 0
);
```

### 4.2.5 MariaDB

```
CREATE TABLE MOOD (
    ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),
    UPDATED_AT  DATETIME,
    NAME_KEY    VARCHAR(100) NOT NULL,
    DESCKEY     VARCHAR(100),
    VALUE       INT UNSIGNED NOT NULL DEFAULT 0
) ENGINE = InnoDB;
```

## 5 Mòdul RSCMOD

### 5.1 Entitat RESOURCE

#### 5.1.1 Proto3

```
message ResResource {
  a_user.usrmod.BaseEntity      base_entity = 1;
  string                        name_key = 2;
  optional string               desc_key = 3;
  int32                         version = 4;
  ResResource                   root_id = 5;
  string                        locale_code = 6;
  b_definitions.resmod.ResResourceType resource_type = 7;
  optional string               inline_text = 8;
  optional string               link = 9;
}
```

#### 5.1.2 Flutter

```
enum ResourceType {
  unspecified,
  plainText,
  richText,
  image,
  audio,
  video,
}

class RscResource extends ModelEntity {
  String? _nameKey;
  String? _name;
  String? _descKey;
  String? _desc;
  int _vers = 0;
  RscResource? _root;
  Locale _locale = localeES;
  ResourceType _type = ResourceType.unspecified;
  String? _inlineText;
  String? _link;
}
```

#### 5.1.3 Golang

PENDENT

#### 5.1.4 SQLite

```
CREATE TABLE $tnRscResource (
  $standardHeader,
  $fldNameKey $dbtTextNotNull,
  $fldDescKey $dbtTextNotNull,
  $fldVersion $dbtIntNotNull,
  $fldRoot $dbtInt REFERENCES $tnRscResource($fldId),
  $fldLocaleCode $dbtTextNotNull,
  $fldResourceType $dbtIntNotNull,
  $fldInlineText $dbtText,
  $fldLink $dbtText
);
```

### 5.1.5 MariaDB

```
CREATE TABLE RESOURCE (
  ID                INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY        INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT        DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY        INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT        DATETIME,
  NAME_KEY          VARCHAR(100) NOT NULL,
  DESC_KEY          VARCHAR(100),
  VERSION           INT UNSIGNED NOT NULL,
  ROOT_ID           INT UNSIGNED REFERENCES RESOURCE(ID),
  LOCALE_CODE       VARCHAR(2) NOT NULL,
  RESOURCE_TYPE     INT UNSIGNED NOT NULL,
  INLINE_TEXT       VARCHAR(10000),
  LINK              VARCHAR(4096)
) ENGINE = InnoDB;
```

## 5.2 Entitat PHASE\_RESOURCE

### 5.2.1 Proto3

```
message ResPhaseResource {
  a_user.usrmod.BaseEntity      base_entity = 1;
  b_definitions.dismod.DisDisease disease = 2;
  optional b_definitions.dismod.DisPhase phase = 3;
  ResResource                   resource = 4;
}
```

### 5.2.2 Flutter

```
class RscPhaseResource extends ModelEntity {
  DisDisease? _disease;
  DisPhase?   _phase;
  RscResource? _resource;
}
```

### 5.2.3 Golang

### 5.2.4 SQLite

```
CREATE TABLE $tnRscPhaseResource (
  $standardHeader,
  $fldDisease      $dbtIntNotNull REFERENCES $tnDisDisease($fldId),
  $fldDiseasePhase $dbtInt REFERENCES $tnDisPhase($fldId),
  $fldResource     $dbtIntNotNull REFERENCES $tnRscResource($fldId)
);
```

### 5.2.5 MariaDB

```
CREATE TABLE PHASE_RESOURCE (
  ID                INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY        INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT        DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY        INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT        DATETIME,
  DISEASE_ID        INT UNSIGNED NOT NULL REFERENCES DISEASE(ID),
  PHASE_ID          INT UNSIGNED REFERENCES PHASE(ID),
  RESOURCE_ID       INT UNSIGNED NOT NULL REFERENCES RESOURCE(ID)
) ENGINE = InnoDB;
```

## 6 Mòdul TCKMOD

### 6.1 Entitat TRACKING

#### 6.1.1 Proto3

```
message TckTracking {
    BaseEntity          base_entity = 1;
    string              name_key = 2;
    optional string     desc_key = 3;
    int32               version = 4;
    UsrUser             therapist = 5;
    optional TckTracking root = 6;
}
```

#### 6.1.2 Flutter

```
class TckTracking extends ModelEntity {
    String? _nameKey;
    String? _name;
    String? _descKey;
    String? _desc;
    int _vers = 0;
    UsrUser? _therapist;
    TckTracking? _root;
}
```

#### 6.1.3 Golang

PENDENT

#### 6.1.4 SQLite

```
CREATE TABLE $tnTckTracking (
    $standardHeader,
    $fldNameKey $dbtTextNotNull,
    $fldDescKey $dbtTextNotNull,
    $fldVersion $dbtIntNotNull DEFAULT 0,
    $fldTherapist $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
    $fldRoot $dbtIntNotNull REFERENCES $tnTckTracking($fldId),
);
```

#### 6.1.5 MariaDB

```
CREATE TABLE TRACKING (
    ID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY INT UNSIGNED REFERENCES USER(ID),
    UPDATED_AT DATETIME,
    NAME_KEY VARCHAR(100) NOT NULL,
    DESC_KEY VARCHAR(100),
    VERSION INT UNSIGNED NOT NULL,
    THERAPIST_ID INT UNSIGNED NOT NULL REFERENCES USER(ID),
    ROOT_ID INT UNSIGNED REFERENCES TRACKING(ID)
) ENGINE = InnoDB;
```

## 6.2 Entitat PHASE\_TRACKING

### 6.2.1 Proto3

```
message TckPhaseTracking {
    BaseEntity base_entity = 1;
    DisPhase phase = 2;
    TckTracking tracking = 3;
}
```

### 6.2.2 Flutter

```
class TckPhaseTracking extends ModelEntity {
    DisPhase? _phase;
    TckTracking? _tracking;
}
```

### 6.2.3 Golang

#### PENDENT

### 6.2.4 SQLite

```
CREATE TABLE $tnTckPhaseTracking (
    $standardHeader,
    $fldDiseasePhase $dbtIntNotNull REFERENCES $tnDisPhase($fldId),
    $fldTracking $dbtIntNotNull REFERENCES $tnTckTracking($fldId)
);
```

### 6.2.5 MariaDB

```
CREATE TABLE PHASE_TRACKING (
    ID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY INT UNSIGNED REFERENCES USER(ID),
    UPDATED_AT DATETIME,
    PHASE_ID INT UNSIGNED NOT NULL REFERENCES PHASE(ID),
    TRACKING_ID INT UNSIGNED NOT NULL REFERENCES TRACKING(ID)
) ENGINE = InnoDB;
```

## 6.3 Entitat TRACKING\_COLUMN

### 6.3.1 Proto3

```
message TckTrackingColumn {
    BaseEntity base_entity = 1;
    TckTracking tracking = 2;
    optional LstOptionList list = 3;
    string title_key = 4;
    TckTrackingColumnType track_column_type = 5;
    bool mandatory = 6;
    bool custom = 7;
}
```



### 6.3.2 Flutter

```
enum TrackingColumnType {
    unspecified,
    plainText,
    boolean,
    list,
    emotion,
    mood,
    date,
    time,
    dateTime,
}

class TckTrackingColumn extends ModelEntity {
    TckTracking?      _tracking;
    LstOptionList?    _list;
    String?           _titleKey;
    String?           _title;
    TrackingColumnType _type = TrackingColumnType.unspecified;
    bool              _mandatory = false;
    bool              _custom = false;
}
```

### 6.3.3 Golang

PENDENT

### 6.3.4 SQLite

```
CREATE TABLE $tnTckTrackingColumn (
    $standardHeader,
    $fldTracking      $dbtIntNotNull REFERENCES $tnTckTracking($fldId),
    $fldList           $dbtInt REFERENCES $tnLstOptionList($fldId),
    $fldTitleKey       $dbtTextNotNull,
    $fldTrackingColumnType $dbtIntNotNull,
    $fldMandatory      $dbtBooleanNotNull,
    $fldCustom         $dbtBooleanNotNull
);
```

### 6.3.5 MariaDB

```
CREATE TABLE TRACKING_COLUMN (
    ID            INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY    INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT    DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY    INT UNSIGNED REFERENCES USER(ID),
    UPDATED_AT    DATETIME,
    TRACKING_ID   INT UNSIGNED NOT NULL REFERENCES TRACKING(ID),
    LIST_ID       INT UNSIGNED REFERENCES OPTION_LIST(ID),
    TITLE         VARCHAR(100) NOT NULL,
    COLUMN_TYPE   INT UNSIGNED NOT NULL,
    MANDATORY     BOOLEAN NOT NULL DEFAULT TRUE,
    CUSTOMIZED     BOOLEAN NOT NULL DEFAULT FALSE
) ENGINE = InnoDB;
```

## 7 Mòdul TSTMOD

### 7.1 Entitat TEST\_CATEGORY

#### 7.1.1 Proto3

```
message TstTestCategory {
    BaseEntity          base_entity = 1;
    string              title_key = 2;
    optional string     desc_key = 3;
    optional TstTestCategory parent = 4;
}
```

#### 7.1.2 Flutter

```
class TstTestCategory extends ModelEntity {
    String? _titleKey;
    String? _title;
    String? _descKey;
    String? _desc;
    TstTestCategory? _parent;
}
```

#### 7.1.3 Golang

#### PENDENT

#### 7.1.4 SQLite

```
CREATE TABLE $tnTstTestCategory (
    $standardHeader,
    $fldTitleKey $dbtTextNotNullUnique,
    $fldDescKey $dbtText,
    $fldParent $dbtInt REFERENCES $tnTstTestCategory($fldId)
);
```

#### 7.1.5 MariaDB

```
CREATE TABLE TEST_CATEGORY (
    ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY INT UNSIGNED REFERENCES USER(ID),
    UPDATED_AT  DATETIME,
    TITLE_KEY   VARCHAR(100) NOT NULL,
    DESC_KEY    VARCHAR(100),
    PARENT_ID   INT UNSIGNED REFERENCES TEST_CATEGORY(ID)
) ENGINE = InnoDB;
```

### 7.2 Entitat TEST

#### 7.2.1 Proto3

```
message TstTest {
    BaseEntity          base_entity = 1;
    TstTestCategory     category = 2;
    string              name_key = 3;
    optional string     desc_key = 4;
    optional string     instr_key = 5;
    int32               version = 6;
}
```

```

    UsrUser          therapist = 7;
    optional TstTest root = 8;
}

```

### 7.2.2 Flutter

```

class TstTest extends ModelEntity {
  stTestCategory? _category;
  String?         _nameKey;
  String?         _name;
  String?         _descKey;
  String?         _desc;
  String?         _instrKey;
  String?         _instr;
  int             _vers = 0;
  UsrUser?        _therapist;
  TstTest?        _root;
}

```

### 7.2.3 Golang

PENDENT

### 7.2.4 SQLite

```

CREATE TABLE $tnTstTest (
  $standardHeader,
  $fldTestCategory $dbtIntNotNull REFERENCES $tnTstTestCategory($fldId),
  $fldNameKey      $dbtTextNotNull,
  $fldDescKey      $dbtText,
  $fldInstrKey     $dbtText,
  $fldVersion      $dbtIntNotNull,
  $fldRoot         $dbtInt REFERENCES $tnTstTest($fldId),
  $fldTherapist    $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
);

```

### 7.2.5 MariaDB

```

CREATE TABLE TEST (
  ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT  DATETIME,
  CATEGORY_ID INT UNSIGNED NOT NULL REFERENCES TEST_CATEGORY(ID),
  NAME_KEY    VARCHAR(100) NOT NULL,
  DESC_KEY    VARCHAR(100),
  INSTRS_KEY  VARCHAR(100),
  VERSION     INT UNSIGNED NOT NULL,
  ROOT_ID     INT UNSIGNED REFERENCES TEST(ID),
  THERAPIST_ID INT UNSIGNED NOT NULL REFERENCES USER(ID)
) ENGINE InnoDB;

```

## 7.3 Entitat TEST\_QUESTION

### 7.3.1 Proto3

```
message TstQuestion {
  BaseEntity          base_entity = 1;
  TstTest             test = 2;
  TstTestQuestionType test_question_type = 3;
  optional TstQuestion block = 4;
  string              question = 5;
  optional string      help = 6;
  bool                mandatory = 7;
  bool                custom = 8;
}
```

### 7.3.2 Flutter

```
enum QuestionType {
  unspecified,
  plainText,
  boolean,
  list,
  emotion,
  mood,
  date,
  dateTime,
}

class TstQuestion extends ModelEntity {
  TstTest? _test;
  QuestionType _type = QuestionType.unspecified;
  TstQuestion? _block;
  String? _questionKey;
  String? _question;
  String? _helpKey;
  String? _help;
  bool _mandatory = false;
  bool _custom = false;
}
```

### 7.3.3 Golang

PENDENT

### 7.3.4 SQLite

```
CREATE TABLE $tnTstQuestion (
  $standardHeader,
  $fldTest          $dbtIntNotNull REFERENCES $tnTstTest($fldId),
  $fldQuestionType  $dbtIntNotNull,
  $fldBlock         $dbtInt REFERENCES $tnTstTest($fldId),
  $fldQuestionKey   $dbtTextNotNull,
  $fldHelpKey       $dbtText,
  $fldMandatory     $dbtBooleanNotNull DEFAULT FALSE,
  $fldCustom        $dbtBooleanNotNull DEFAULT FALSE
);
```

### 7.3.5 MariaDB

```
CREATE TABLE QUESTION (  
  ID                INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  CREATED_BY        INT UNSIGNED NOT NULL REFERENCES USER(ID),  
  CREATED_AT        DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  UPDATED_BY        INT UNSIGNED REFERENCES USER(ID),  
  UPDATED_AT        DATETIME,  
  TEST_ID           INT UNSIGNED NOT NULL REFERENCES TEST(ID),  
  QUESTION_TYPE     INT UNSIGNED NOT NULL,  
  BLOCK_ID          INT UNSIGNED REFERENCES QUESTION(ID),  
  QUESTION_KEY      VARCHAR(100) NOT NULL,  
  HELP_KEY          VARCHAR(100),  
  MANDATORY         BOOLEAN NOT NULL DEFAULT TRUE,  
  CUSTOMIZED        BOOLEAN NOT NULL DEFAULT FALSE  
) ENGINE = InnoDB;
```

## 8 Mòdul DGNMOD

### 8.1 Entitat DIAGNOSIS

#### 8.1.1 Proto3

```
message DgnDiagnosis {
  a_user.usrmod.BaseEntity      base_entity = 1;
  b_definitions.dismod.DisDisease disease = 2;
  a_user.usrmod.UsrUser         therapist = 3;
  c_working.dgnmod.DgnDiagnosisState diagnosis_state = 4;
  optional string                annotations = 5;
}
```

#### 8.1.2 Flutter

```
enum DiagnosisState {
  unspecified,
  stablished,
  cancelled,
  cured,
  chronicled,
}

class DgnDiagnosis extends ModelEntity {
  DisDisease? _disease;
  UsrUser?    _therapist;
  DiagnosisState _state = DiagnosisState.unspecified;
  String?      _annotations;
}
```

#### 8.1.3 Golang

#### PENDENT

#### 8.1.4 SQLite

```
CREATE TABLE $tnDgnDiagnosis (
  $standardHeader,
  $fldDisease      $dbtIntNotNull REFERENCES $tnDisDisease($fldId),
  $fldTherapist    $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
  $fldDiagnosisState $dbtIntNotNull DEFAULT 0,
  $fldAnnotations  $dbtText
);
```

#### 8.1.5 MariaDB

```
CREATE TABLE DIAGNOSIS (
  ID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT DATETIME,
  DISEASE_ID INT UNSIGNED NOT NULL REFERENCES DISEASE(ID),
  THERAPIST_ID INT UNSIGNED NOT NULL REFERENCES USER(ID),
  DIAGNOSIS_STATE INT UNSIGNED NOT NULL,
  ANNOTATIONS VARCHAR(2048)
) ENGINE = InnoDB;
```

## 8.2 Entitat DIAGNOSIS\_PHASE

### 8.2.1 Proto3

```
message DgnDiagnosisPhase {
  a_user.usrmod.BaseEntity      base_entity = 1;
  DgnDiagnosis                  diagnosis = 2;
  b_definitions.dismod.DisPhase disease_phase = 3;
  d_admin.vismod.VisVisit       visit = 4;
  c_working.dgnmod.DgnDiagnosisPhaseState state = 5;
  optional DgnAchievement       relapse = 6;
  optional string                annotations = 7;
}
```

### 8.2.2 Flutter

```
enum DiagnosisPhaseState {
  unspecified,
  opened,
  achieved,
  relapsed,
  chronicled,
}

class DgnDiagnosisPhase extends ModelEntity {
  DgnDiagnosis?      _diagnosis;
  DisPhase?          _phase;
  VisVisit?          _visit;
  DiagnosisPhaseState _state = DiagnosisPhaseState.unspecified;
  DgnDiagnosisPhase? _relapse;
  String?             _annotations;
}
```

### 8.2.3 Golang

#### PENDENT

### 8.2.4 SQLite

```
CREATE TABLE $tnDgnDiagnosisPhase (
  $standardHeader,
  $fldDiagnosis          $dbtIntNotNull REFERENCES $tnDgnDiagnosis($fldId),
  $fldDiseasePhase       $dbtIntNotNull REFERENCES $tnDisPhase($fldId),
  $fldVisit              $dbtIntNotNull REFERENCES $tnVisVisit($fldId),
  $fldDiagnosisPhaseState $dbtIntNotNull DEFAULT 0,
  $fldRelapse            $dbtInt REFERENCES $tnDgnDiagnosisPhase($fldId),
  $fldAnnotations        $dbtText
);
```

### 8.2.5 MariaDB

```
CREATE TABLE DIAGNOSIS_PHASE (
  ID                INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY        INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT        DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY        INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT        DATETIME,
  DIAGNOSIS_ID      INT UNSIGNED NOT NULL REFERENCES DIAGNOSIS(ID),
  DISEASE_PHASE_ID  INT UNSIGNED NOT NULL REFERENCES PHASE(ID),
  VISIT_ID          INT UNSIGNED NOT NULL REFERENCES VISIT(ID),
```

```

    PHASE_STATE      INT UNSIGNED NOT NULL,
    RELAPSE_ID       INT UNSIGNED REFERENCES DIAGNOSIS_PHASE(ID),
    ANNOTATIONS      VARCHAR(2048)
) ENGINE = InnoDB;

```

## 8.3 Entitat ACHIEVEMENT

### 8.3.1 Proto3

```

message DgnAchievement {
    a_user.usrmod.BaseEntity      base_entity = 1;
    b_definitions.dismod.DisGoal   goal = 2;
    DgnDiagnosisPhase             phase = 3;
    c_working.dgnmod.DgnDiagnosisPhaseState state = 4;
    optional DgnAchievement        relapse = 5;
    optional string                 annotations = 6;
}

```

### 8.3.2 Flutter

```

enum AchievementState {
    unspecified,
    proposed,
    pending,
    cancelled,
    achieved,
    relapsed,
}

class DgnAchievement extends ModelEntity {
    DisGoal?      _goal;
    DgnDiagnosisPhase? _phase;
    AchievementState _state = AchievementState.unspecified;
    DgnAchievement? _relapse;
    String?        _annotations;
}

```

### 8.3.3 Golang

#### PENDENT

### 8.3.4 SQLite

```

CREATE TABLE $tnDgnAchievement (
    $standardHeader,
    $fldGoal          $dbtIntNotNull REFERENCES $tnDisGoal($fldId),
    $fldDiagnosisPhase $dbtIntNotNull REFERENCES $tnDgnDiagnosisPhase($fldId),
    $fldRelapse       $dbtInt REFERENCES $tnDgnAchievement($fldId),
    $fldAchievementState $dbtIntNotNull DEFAULT 0,
    $fldAnnotations   $dbtText
);

```



### 8.3.5 MariaDB

```
CREATE TABLE ACHIEVEMENT (  
  ID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  CREATED_BY INT UNSIGNED NOT NULL REFERENCES USER(ID),  
  CREATED_AT DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  UPDATED_BY INT UNSIGNED REFERENCES USER(ID),  
  UPDATED_AT DATETIME,  
  GOAL_ID INT UNSIGNED NOT NULL REFERENCES GOAL(ID),  
  DIAGNOSIS_PHASE_ID INT UNSIGNED NOT NULL REFERENCES DIAGNOSIS_PHASE(ID),  
  ACV_STATE INT UNSIGNED NOT NULL,  
  RELAPSE_ID INT UNSIGNED REFERENCES ACHIEVEMENT(ID),  
  ANNOTATIONS VARCHAR(2048)  
) ENGINE = InnoDB;
```

## 9 Mòdul MATMOD

### 9.1 Entitat MATERIAL

#### 9.1.1 Proto3

```
message MatMaterial {
  a_user.usrmod.BaseEntity      base_entity = 1;
  e_localization.locmod.string   name = 2;
  optional e_localization.locmod.string desc_key = 3;
  b_definitions.resmod.ResResource resource = 4;
  b_definitions.resmod.ResResourceType resource_type = 5;
  e_localization.locmod.string   containt = 6;
  string                         link = 7;
}
```

#### 9.1.2 Flutter

```
class MatMaterial extends ModelEntity {
  String? _nameKey;
  String? _name;
  String? _descKey;
  String? _desc;
  RscResource? _resource;
  ResourceType _type = ResourceType.unspecified;
  String? _plainKey;
  String? _plain;
  String? _link;
}
```

#### 9.1.3 Golang

#### 9.1.4 SQLite

```
CREATE TABLE $tnMatMaterial (
  $standardHeader,
  $fldNameKey      $dbtTextNotNullUnique,
  $fldDescKey      $dbtText,
  $fldResource      $dbtIntNotNull REFERENCES $tnRscResource($fldId),
  $fldResourceType $dbtIntNotNull DEFAULT 0,
  $fldPlainKey      $dbtText,
  $fldLink          $dbtText
);
```

#### 9.1.5 MariaDB

```
CREATE TABLE MATERIAL (
  ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT  DATETIME,
  NAME_KEY    VARCHAR(100) NOT NULL,
  DESC_KEY    VARCHAR(100),
  RESOURCE_ID INT UNSIGNED NOT NULL REFERENCES RESOURCE(ID),
  MATERIAL_TYPE INT UNSIGNED NOT NULL,
  PLAIN_KEY    VARCHAR(100),
  LINK        VARCHAR(4096)
) ENGINE = InnoDB;
```

## 9.2 Entitat MATERIAL\_PHASE

### 9.2.1 Proto3

```
message MatPhaseMaterial {
  a_user.usrmod.BaseEntity      base_entity = 1;
  b_definitions.resmod.ResPhaseResource phase = 2;
  a_user.usrmod.UsrUser          patient = 3;
}
```

### 9.2.2 Flutter

```
class MatPhaseMaterial extends ModelEntity {
  UsrUser?      _patient;
  RscPhaseResource? _phase;
}
```

### 9.2.3 Golang

PENDENT

### 9.2.4 SQLite

```
CREATE TABLE $tnMatPhaseMaterial (
  $standardHeader,
  $fldPatient      $dbtIntNotNull REFERENCES $tnUsrUser($fldId)
  $fldPhaseResource $dbtIntNotNull REFERENCES $tnRscPhaseResource($fldId)
);
```

### 9.2.5 MariaDB

```
CREATE TABLE PHASE_MATERIAL (
  ID                INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY        INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT        DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY        INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT        DATETIME,
  PATIENT_ID        INT UNSIGNED NOT NULL REFERENCES USER(ID),
  PHASE_RESOURCE_ID INT UNSIGNED NOT NULL REFERENCES PHASE_RESOURCE(ID)
) ENGINE InnoDB;
```

## 10 Mòdul REGMOD

### 10.1 Entitat REGISTER

#### 10.1.1 Proto3

```
message RegRegister {
  a_user.usrmod.BaseEntity      base_entity = 1;
  b_definitions.tckmod.TckTracking tracking = 2;
  a_user.usrmod.UsrUser          patient = 3;
  optional Timestamp             first_date = 4;
  optional Timestamp             last_date = 5;
  c_working.regmod.RegRegisterState state = 6;
}
```

#### 10.1.2 Flutter

```
enum RegisterState {
  unspecified,
  assigned,
  statical,
  cancelled,
}
```

```
class RegRegister extends ModelEntity {
  TckTracking? _tracking;
  UsrUser?     _patient;
  DateTime?    _firstDate;
  DateTime?    _lastDate;
  RegisterState _state = RegisterState.unspecified;
}
```

#### 10.1.3 Golang

#### 10.1.4 SQLite

```
CREATE TABLE $tnRegRegister (
  $standardHeader,
  $fldTracking      $dbtIntNotNull REFERENCES $tnTckTracking($fldId),
  $fldPatient       $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
  $fldFirstDate     $dbtDateTime,
  $fldLastDate      $dbtDateTime,
  $fldRegisterState $dbtIntNotNull DEFAULT 0
);
```

#### 10.1.5 MariaDB

```
CREATE TABLE REGISTER (
  ID                INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY       INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT       DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY       INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT       DATETIME,
  TRACKING_ID      INT UNSIGNED NOT NULL REFERENCES TRACKING(ID),
  PATIENT_ID       INT UNSIGNED NOT NULL REFERENCES USER(ID),
  FIRST_DATETIME   DATETIME,
  LAST_DATETIME    DATETIME,
  REGISTER_STATE   INT UNSIGNED NOT NULL
) ENGINE = InnoDB;
```

## 10.2 Entitat REGISTER\_COLUMN

### 10.2.1 Proto3

```
message RegRegisterColumn {
  a_user.usrmod.BaseEntity          base_entity = 1;
  RegRegister                        register = 2;
  b_definitions.tckmod.TckTrackingColumn column = 3;
  optional e_localization.lstmod.LstOptionEntry option = 4;
  optional b_definitions.emomod.EmoEmotion emotion = 5;
  optional b_definitions.emomod.EmoMood mood = 6;
  optional string                    value = 7;
}
```

### 10.2.2 Flutter

```
class RegRegisterColumn extends ModelEntity {
  RegRegister? _register;
  TckTrackingColumn? _column;
  LstOptionEntry? _option;
  EmoEmotion? _emotion;
  EmoMood? _mood;
  String? _value;
}
```

### 10.2.3 Golang

#### PENDENT

### 10.2.4 SQLite

```
CREATE TABLE $tnRegRegisterColumn (
  $standardHeader,
  $fldRegister          $dbtIntNotNull REFERENCES $tnRegRegister($fldId),
  $fldTrackingColumn    $dbtIntNotNull REFERENCES $tnTckTrackingColumn($fldId),
  $fldOptionEntry       $dbtInt REFERENCES $tnLstOptionEntry($fldId),
  $fldEmotion           $dbtInt REFERENCES $tnEmoEmotion($fldId),
  $fldMood              $dbtInt REFERENCES $tnEmoMood($fldId),
  $fldValue             $dbtInt
);
```

### 10.2.5 MariaDB

```
CREATE TABLE REGISTER_COLUMN (
  ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT  DATETIME,
  REGISTER_ID INT UNSIGNED NOT NULL REFERENCES REGISTER(ID),
  TRACK_COL_ID INT UNSIGNED NOT NULL REFERENCES TRACKING_COLUMN(ID),
  OPTION_ID   INT UNSIGNED REFERENCES OPTION_LIST(ID),
  EMOTION_ID  INT UNSIGNED REFERENCES EMOTION(ID),
  MOOD_ID     INT UNSIGNED REFERENCES MOOD(ID),
  VALUE       VARCHAR(4096)
) ENGINE = InnoDB;
```

# 11 Mòdul RESMOD

## 11.1 Entitat PATIENT\_TEST

### 11.1.1 Proto3

```
message ResPatientTest {
  BaseEntity          base_entity = 1;
  TstTest             test = 2;
  UsrUser             therapist = 3;
  UsrUser             patient = 4;
  optional Timestamp  assign_date = 5;
  optional Timestamp  finish_date = 6;
  optional UsrUser    revised_by = 7;
  optional Timestamp  revision_date = 8;
  optional string     valuation = 9;
}
```

### 11.1.2 Flutter

```
class ResPatientTest extends ModelEntity {
  UsrUser? _therapist; // NOT NULL
  UsrUser? _patient; // NOT NULL
  DateTime? _assignAt;
  DateTime? _completedAt;
  UsrUser? _evaluatedBy;
  DateTime? _evaluatedAt;
  String? _evaluation;
}
```

### 11.1.3 Golang

### 11.1.4 SQLite

```
CREATE TABLE $tnResPatientTest (
  $standardHeader,
  $fldTherapist $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
  $fldPatient $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
  $fldAssignedAt $dbtDateTime,
  $fldCompletedAt $dbtDateTime,
  $fldEvaluatedBy $dbtInt REFERENCES $tnUsrUser($fldId),
  $fldEvaluatedAt $dbtDateTime,
  $fldEvaluation $dbtText
);
```

### 11.1.5 MariaDB

```
CREATE TABLE PATIENT_TEST (
  ID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT DATETIME,
  THERAPIST_ID INT UNSIGNED NOT NULL REFERENCES USER(ID),
  PATIENT_ID INT UNSIGNED NOT NULL REFERENCES USER(ID),
  ASSIGNED_AT DATETIME,
  COMPLETED_AT DATETIME,
  EVALUATED_BY INT UNSIGNED REFERENCES USER(ID),
  EVALUATED_AT DATETIME,
  EVALUATION VARCHAR(4096)
```

```
) ENGINE = InnoDB;
```

## 11.2 Entitat ANSWER

### 11.2.1 Proto3

```
message ResAnswer {
    BaseEntity          base_entity = 1;
    ResPatientTest      test = 2;
    TstQuestion         question = 3;
    optional string      answer = 4;
    optional LstOptionEntry option = 5;
    optional EmoEmotion  emotion = 6;
    optional EmoMood     mood = 7;
    optional Timestamp   valued_date = 8;
    optional UsrUser     revised_by = 9;
    optional Timestamp   revision_date = 10;
    optional string      valoration = 11;
}
```

### 11.2.2 Flutter

```
class ResAnswer extends ModelEntity {
    TstTest?      _test;
    TstQuestion?  _question;
    String?       _answer;
    LstOptionList? _list;
    EmoEmotion?   _emotion;
    EmoMood?      _mood;
    DateTime?     _completedAt;
    UsrUser?      _evaluatedBy;
    DateTime?     _evaluatedAt;
    String?       _evaluation;
}
```

### 11.2.3 Golang

#### PENDENT

### 11.2.4 SQLite

```
CREATE TABLE $tnResAnswer (
    $standardHeader,
    $fldTest          $dbtIntNotNull REFERENCES $tnTstTest($fldId),
    $fldQuestionId    $dbtIntNotNull REFERENCES $tnTstQuestion($fldId),
    $fldAnswer        $dbtText,
    $fldList          $dbtIntNotNull REFERENCES $tnLstOptionList($fldId),
    $fldEmotion       $dbtIntNotNull REFERENCES $tnEmoEmotion($fldId),
    $fldMood          $dbtIntNotNull REFERENCES $tnEmoMood($fldId),
    $fldCompletedAt   $dbtDateTime,
    $fldEvaluatedBy   $dbtInt REFERENCES $tnUsrUser($fldId),
    $fldEvaluatedAt   $dbtDateTime,
    $fldEvaluation     $dbtText
);
```

### 11.2.5 MariaDB

```
CREATE TABLE ANSWER (  
  ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),  
  CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),  
  UPDATED_AT  DATETIME,  
  TEST_ID     INT UNSIGNED NOT NULL REFERENCES PATIENT_TEST(ID),  
  QUESTION_ID INT UNSIGNED NOT NULL REFERENCES QUESTION(ID),  
  ANSWER      VARCHAR(2048),  
  LIST_ID     INT UNSIGNED REFERENCES OPTION_LIST(ID),  
  EMOTION_ID  INT UNSIGNED REFERENCES EMOTION(ID),  
  MOOD_ID     INT UNSIGNED REFERENCES MOOD(ID),  
  COMPLETED_AT DATETIME,  
  EVALUATED_BY INT UNSIGNED REFERENCES USER(ID),  
  EVALUATED_AT DATETIME,  
  EVALUATION  VARCHAR(4096)  
) ENGINE = InnoDB;
```



## 12 Mòdul NTFMOD

### 12.1 Entitat NOTIFICATION

#### 12.1.1 Proto3

```
message NtfNotification {
  a_user.usrmod.BaseEntity      base_entity = 1;
  a_user.usrmod.UsrUser         user = 2;
  string                        notification = 3;
  d_admin.ntfmod.NtfNotificationType notification_type = 4;
  d_admin.ntfmod.NtfNotificationState notification_state = 5;
  optional c_working.resmod.ResPatientTest test = 6;
  optional c_working.regmod.RegRegister tracking = 7;
  optional c_working.matmod.MatMaterial material = 8;
}
```

#### 12.1.2 Flutter

```
enum NotificationType {
  unspecified,
  type1,
  type2,
  type3,
}

enum NotificationState {
  unspecified,
  received,
  readed,
}
```

```
class NtfNotification extends ModelEntity {
  UsrUser?      _user;
  String?       _notification;
  NotificationType _type = NotificationType.unspecified;
  NotificationState _state = NotificationState.unspecified;
  TstTest?      _test;
  RegRegister?  _register;
  RscResource?  _resource;
}
```

#### 12.1.3 Golang

#### PENDENT

#### 12.1.4 SQLite

```
CREATE TABLE $tnNtfNotification (
  $standardHeader,
  $fldUser          $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
  $fldNotification  $dbtTextNotNull,
  $fldNotificationType $dbtIntNotNull DEFAULT 0,
  $fldNotificationState $dbtIntNotNull DEFAULT 0,
  $fldTest          $dbtInt REFERENCES $tnTstTest($fldId),
  $fldRegister      $dbtInt REFERENCES $tnRegRegister($fldId),
  $fldResource      $dbtInt REFERENCES $tnRscResource($fldId)
);
```

### 12.1.5 MariaDB

```
CREATE TABLE NOTIFICATION (  
  ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),  
  CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),  
  UPDATED_AT  DATETIME,  
  USER_ID     INT UNSIGNED NOT NULL REFERENCES USER(ID),  
  NOTIFICATION VARCHAR(1024) NOT NULL,  
  NTF_TYPE    INT UNSIGNED NOT NULL,  
  NTF_STATE   INT UNSIGNED NOT NULL,  
  TEST_ID     INT UNSIGNED REFERENCES PATIENT_TEST(ID),  
  REGISTER_ID INT UNSIGNED REFERENCES REGISTER(ID),  
  RESOURCE_ID INT UNSIGNED REFERENCES RESOURCE(ID)  
) ENGINE = InnoDB;
```

## 13 Mòdul TSKMOD

### 13.1 Entitat TASK

#### 13.1.1 Proto3

```
message TskTask {
  a_user.usrmod.BaseEntity          base_entity = 1;
  a_user.usrmod.UsrUser              therapist = 2;
  a_user.usrmod.UsrUser              patient = 3;
  d_admin.tskmod.TskTaskType         task_type = 4;
  d_admin.tskmod.TskTaskState        task_state = 5;
  string                             task = 6;
  optional c_working.resmod.ResPatientTest test = 7;
  optional c_working.regmod.RegRegister tracking = 8;
  optional c_working.matmod.MatMaterial material = 9;
  optional Timestamp                 start_date = 10;
  optional Timestamp                 end_date = 11;
}
```

#### 13.1.2 Flutter

```
enum TaskType {
  unspecified,
  test,
  register,
  resource,
  custom,
}

enum TaskState {
  unspecified,
  assigned,
  finished,
  statical,
}
```

```
class TskTask extends ModelEntity {
  UsrUser? _therapist;
  UsrUser? _patient;
  TaskType _type = TaskType.unspecified;
  TaskState _state = TaskState.unspecified;
  String? _descKey;
  String? _desc;
  TstTest? _test;
  RegRegister? _register;
  RscResource? _resource;
  DateTime? _startDate;
  DateTime? _endDate;
}
```

#### 13.1.3 Golang

#### PENDENT

#### 13.1.4 SQLite

```
CREATE TABLE $tnTskTask (
  $standardHeader,
  $fldTherapist $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
  $fldPatient $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
  $fldTaskType $dbtIntNotNull DEFAULT 0,
  $fldTaskState $dbtIntNotNull DEFAULT 0,
  $fldDescKey $dbtTextNotNull,
  $fldTest $dbtIntNotNull REFERENCES $tnTstTest($fldId),
```

```

    $fldRegister $dbtIntNotNull REFERENCES $tnRegRegister($fldId),
    $fldResource $dbtIntNotNull REFERENCES $tnRscResource($fldId),
    $fldStartDate $dbtDateTime,
    $fldEndDate $dbtDateTime
);

```

### 13.1.5 MariaDB

```

CREATE TABLE TASK (
  ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT  DATETIME,
  THERAPIST_ID INT UNSIGNED NOT NULL REFERENCES USER(ID),
  PATIENT_ID  INT UNSIGNED NOT NULL REFERENCES USER(ID),
  TASK_TYPE   INT UNSIGNED NOT NULL,
  TASK_STATE  INT UNSIGNED NOT NULL,
  DESC_KEY    VARCHAR(2048) NOT NULL,
  TEST_ID     INT UNSIGNED REFERENCES PATIENT_TEST(ID),
  REGISTER_ID INT UNSIGNED REFERENCES REGISTER(ID),
  RESOURCE_ID INT UNSIGNED REFERENCES RESOURCE(ID),
  START_DATE  DATETIME,
  END_DATE    DATETIME
) ENGINE = InnoDB;

```

## 14 Mòdul VISMODO

### 14.1 Entitat VISIT

#### 14.1.1 Proto3

```
message VisVisit {
  a_user.usrmod.BaseEntity base_entity = 1;
  a_user.usrmod.UsrUser therapist = 2;
  a_user.usrmod.UsrUser patient = 3;
  bool first_visit = 4;
  optional Timestamp date_time = 5;
  d_admin.vismod.VisVisitType visit_type = 6;
  d_admin.vismod.VisVisitState visit_state = 7;
}
```

#### 14.1.2 Flutter

```
enum VisitType {
  unspecified,
  face2face,
  online,
}

enum VisitState {
  unspecified,
  stablished,
  finnished,
  cancelled,
  nulled,
}
```

```
class VisVisit extends ModelEntity {
  UsrUser? _therapist;
  UsrUser? _patient;
  bool _isFirst = false;
  DateTime? _dateTime;
  VisitType _type = VisitType.unspecified;
  VisitState _state = VisitState.unspecified;
}
```

#### 14.1.3 Golang

#### PENDENT

#### 14.1.4 SQLite

```
CREATE TABLE $tnVisVisit (
  $standardHeader,
  $fldTherapist $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
  $fldPatient $dbtIntNotNull REFERENCES $tnUsrUser($fldId),
  $fldIsFirst $dbtBooleanNotNull DEAFALUT 0,
  $fldDateTime $dbtDateTimeNotNull,
  $fldVisitType $dbtIntNotNull DEFAULT 0,
  $fldVisitState $dbtIntNotNull DEFAULT 0
);
```

#### 14.1.5 MariaDB

```
CREATE TABLE VISIT (
  ID INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT DATETIME,
```

```
THERAPIST_ID INT UNSIGNED NOT NULL REFERENCES USER(ID),  
PATIENT_ID   INT UNSIGNED NOT NULL REFERENCES USER(ID),  
IS_FIRST     BOOLEAN NOT NULL DEFAULT FALSE,  
DATE_TIME    DATETIME NOT NULL,  
VISIT_TYPE   INT UNSIGNED NOT NULL,  
VISIT_STATE  INT UNSIGNED NOT NULL  
) ENGINE = InnoDB;
```

## 15 Mòdul LOCMOD

### 15.1 Entitat TRANSLATION

#### 15.1.1 Proto3

```
message LocTranslation {
  string locale_id = 1;
  string text_key = 2;
  string literal = 3;
  int    iteration = 4;
  bool   is_new = 5;
  bool   is_updated = 6;
  bool   is_deleted = 7;
}
```

#### 15.1.2 Flutter

```
class LocTranslation extends ModelEntity {
  late String? _tKey;
  late Locale  _locale = localeES;
  late String? _literal;
  late int?    _iteration;
}
```

#### 15.1.3 Golang

#### PENDENT

#### 15.1.4 SQLite

```
CREATE TABLE $tnLocTranslation (
  $fldLocaleCode $dbtTextNotNull,
  $fldTextKey    $dbtTextNotNull,
  $fldLiteral    $dbtTextNotNull,
  $fldIteration  $dbtIntNotNull DEFAULT 0
);
CREATE UNIQUE INDEX ${tnLocTranslation}_PK ON $tnLocTranslation($fldLocaleCode, $fldTextKey);
```

#### 15.1.5 MariaDB

```
CREATE TABLE TRANSLATION (
  ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  TEXT_KEY    VARCHAR(100) NOT NULL,
  LOCALE_CODE VARCHAR(2) NOT NULL,
  LITERAL     VARCHAR(5000) NOT NULL,
  UNIQUE(TEXT_KEY, LOCALE_CODE),
  FULLTEXT (LITERAL)
) ENGINE = InnoDB;
```

## 16 Mòdul LSTMOD

### 16.1 Entitat LIST\_CATEGORY

#### 16.1.1 Proto3

```
message LstListCategory {
    BaseEntity          base_entity = 1;
    string              name_key = 2;
    optional string     desc_key = 3;
    optional LstListCategory parent = 4;
}
```

#### 16.1.2 Flutter

```
class LstListCategory extends ModelEntity {
    String?      _nameKey;
    String?      _name;
    String?      _descKey;
    String?      _desc;
    LstListCategory? _parent;
}
```

#### 16.1.3 Golang

#### 16.1.4 SQLite

```
CREATE TABLE $tnLstListCategory (
    $standardHeader,
    $fldNameKey $dbtTextNotNullUnique,
    $fldDescKey $dbtText,
    $fldParent $dbtInt REFERENCES $tnLstListCategory($fldId),
);
```

#### 16.1.5 MariaDB

```
CREATE TABLE LIST_CATEGORY (
    ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
    CREATED_BY INT UNSIGNED NOT NULL REFERENCES USER(ID),
    CREATED_AT DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    UPDATED_BY INT UNSIGNED REFERENCES USER(ID),
    UPDATED_AT DATETIME,
    NAME_KEY   VARCHAR(100) NOT NULL,
    DESC_KEY   VARCHAR(100),
    PARENT_ID  INT UNSIGNED REFERENCES LIST_CATEGORY(ID),
    INDEX(PARENT_ID)
) ENGINE = InnoDB;
```

### 16.2 Entitat OPTION\_LIST

#### 16.2.1 Proto3

```
message LstOptionList {
    BaseEntity          base_entity = 1;
    string              name_key = 2;
    optional string     desc_key = 3;
    LstListCategory     category = 4;
    bool                alpha = 5;
}
```



**16.2.2 Flutter**

```
class LstOptionList extends ModelEntity {
  String?      _nameKey;
  String?      _name;
  String?      _descKey;
  String?      _desc;
  LstListCategory? _category;
  bool         _isAlpha = false;
}
```

**16.2.3 Golang****PENDENT****16.2.4 SQLite**

```
CREATE TABLE $tnLstOptionList (
  $standardHeader,
  $fldNameKey      $dbtTextNotNull,
  $fldDescKey      $dbtText,
  $fldListCategory $dbtIntNotNull REFERENCES $tnLstListCategory($fldId),
  $fldIsAlpha      $dbtBooleanNotNull DEFAULT 0
);
```

**16.2.5 MariaDB**

```
CREATE TABLE OPTION_LIST (
  ID                INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY        INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT        DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY        INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT        DATETIME,
  NAME_KEY          VARCHAR(100) NOT NULL,
  DESC_KEY          VARCHAR(100) NOT NULL,
  LIST_CATEGORY_ID  INT UNSIGNED NOT NULL REFERENCES LIST_CATEGORY(ID),
  IS_ALPHA          BOOLEAN NOT NULL DEFAULT FALSE
) ENGINE = InnoDB;
```

## 16.3 Entitat OPTION\_ENTRY

### 16.3.1 Proto3

```
message LstOptionEntry {
  BaseEntity      base_entity = 1;
  LstOptionList   list = 2;
  int32           index = 3;
  string          option_key = 4;
  optional string desc_key_key = 5;
}
```

### 16.3.2 Flutter

```
class LstOptionEntry extends ModelEntity {
  LstOptionList? _list;
  int?           _idx;
  String?        _optionKey;
  String?        _option;
  String?        _descKey;
  String?        _desc;
  bool           _isAlpha = false;
}
```

### 16.3.3 Golang

#### PENDENT

### 16.3.4 SQLite

```
CREATE TABLE $tnLstOptionEntry (
  $standardHeader,
  $fldList      $dbtIntNotNull REFERENCES $tnLstOptionList,
  $fldIdx       $dbtIntNotNull DEFAULT 0,
  $fldOptionKey $dbtTextNotNull,
  $fldDescKey   $dbtTextNotNull,
  $fldIsAlpha   $dbtBooleanNotNull DEFAULT 0
);
```

### 16.3.5 MariaDB

```
CREATE TABLE OPTION_ENTRY (
  ID          INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  CREATED_BY  INT UNSIGNED NOT NULL REFERENCES USER(ID),
  CREATED_AT  DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  UPDATED_BY  INT UNSIGNED REFERENCES USER(ID),
  UPDATED_AT  DATETIME,
  LIST_ID     INT UNSIGNED NOT NULL REFERENCES OPTION_LIST(ID),
  IDX         INT UNSIGNED NOT NULL,
  OPTION_KEY  VARCHAR(100) NOT NULL,
  DESC_KEY    VARCHAR(100),
  IS_ALPHA    BOOLEAN NOT NULL DEFAULT FALSE
) ENGINE = InnoDB;
```

# 17 Índexs

## Diagrames

Diagrama 1: Implementació del Model de Dades.....2

## Entitats

Entitat ACHIEVEMENT.....26

Entitat ANSWER.....33

Entitat DEVICE.....6

Entitat DIAGNOSIS.....24

Entitat DIAGNOSIS\_PHASE.....25

Entitat DISEASE.....9

Entitat DSM\_V.....9

Entitat EMOTION.....13

Entitat FCM\_HISTORY.....7

Entitat GOAL.....11

Entitat LIST\_CATEGORY.....42

Entitat MATERIAL.....28

Entitat MATERIAL\_PHASE.....29

Entitat MOOD.....14

Entitat NOTIFICATION.....35

Entitat OPTION\_ENTRY.....44

Entitat OPTION\_LIST.....42

Entitat PATIENT\_TEST.....32

Entitat PHASE.....10

Entitat PHASE\_RESOURCE.....16

Entitat PHASE\_TRACKING.....18

Entitat REGISTER.....30

Entitat REGISTER\_COLUMN.....31

Entitat RESOURCE.....15

Entitat TASK.....37

Entitat TEST.....20

Entitat TEST\_CATEGORY.....20

Entitat TEST\_QUESTION.....22

Entitat TRACKING.....17

Entitat TRACKING\_COLUMN.....18

Entitat TRANSLATION.....41

Entitat USER.....4

Entitat VISIT.....39