

```
// schema.prisma
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL") // Configurado para Supabase
}

model User {
  id          String  @id @default(uuid())
  email       String  @unique
  passwordHash String
  role        UserRole @default(CLIENT)
  profile     Profile?
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
}

enum UserRole {
  CLIENT
  CAREGIVER
  ADMIN
}

model Profile {
  id          String  @id @default(uuid())
```

```
userId    String    @unique
user      User      @relation(fields: [userId], references: [id])
fullName  String
phone     String
address   Address?
bio       String?
avatarUrl String?
documents Document[]
createdAt DateTime @default(now())
updatedAt DateTime @updatedAt
}
```

```
model CaregiverProfile {
  id          String    @id @default(uuid())
  profileId   String    @unique
  profile     Profile   @relation(fields: [profileId], references: [id])
  specialties  Specialty[]
  certifications Certification[]
  experienceYears Int
  hourlyRate   Float
  availability  Availability[]
  reviews      Review[]
}
```

```
model ClientProfile {
  id      String    @id @default(uuid())
  profileId String    @unique
  profile  Profile   @relation(fields: [profileId], references: [id])
}
```

```
elderInfo ElderInfo[]
requests ServiceRequest[]
}
```

```
model ElderInfo {
  id      String  @id @default(uuid())
  clientId String
  client   ClientProfile @relation(fields: [clientId], references: [id])
  name     String
  age      Int
  conditions String[]
  needs    String[]
}
```

```
model ServiceRequest {
  id      String  @id @default(uuid())
  clientId String
  client   ClientProfile @relation(fields: [clientId], references: [id])
  title    String
  description String
  schedule Schedule
  status    RequestStatus @default(PENDING)
  proposals Proposal[]
  createdAt DateTime @default(now())
}
```

```
model Proposal {
  id      String  @id @default(uuid())
```

```

requestId String
request ServiceRequest @relation(fields: [requestId], references: [id])
caregiverId String
caregiver CaregiverProfile @relation(fields: [caregiverId], references: [id])
message String
price Float
status ProposalStatus @default(SENT)
contract Contract?
createdAt DateTime @default(now())
}

```

```

model Contract {
  id String @id @default(uuid())
  proposalId String @unique
  proposal Proposal @relation(fields: [proposalId], references: [id])
  startDate DateTime
  endDate DateTime?
  terms String
  status ContractStatus @default(ACTIVE)
  payments Payment[]
  createdAt DateTime @default(now())
}

```

// Outros modelos (Address, Document, Specialty, Certification, Availability, Review, Schedule, Payment) omitidos por brevidade

// Enums (RequestStatus, ProposalStatus, ContractStatus) também omitidos