

Projecte cr dito de s ntesis

Creaci  d'una app

Natalia Suarez Benitez

2 de septiembre del 2020

Índex

Índex	1
Introducció	2
NF1:Planificació	3
Estudi de mercat	3
Planificació	4
Pressupost materials i recursos	6
Distribució de l'aplicació i màrqueting	7
DAFO	8
NF2:Disseny del projecte	9
Anàlisi funcional	9
Diagrama de casos d'ús	13
Wireframes	14
Diagrames de sistemes de fitxers	18
Model E/R i relacional de la base de dades	19
NF3:Implementació	21
Codi implementat	21
Conclusions	35



Introducció

Aquest és el projecte de credit de síntesis de segon any de grau superior de desenvolupament d'aplicacions multiplataforma, el qual consisteix en desenvolupar un programa o aplicació amb una base de dades amb els seus corresponents models relacionals i entitat relació, és necessari que inclou diagrames de classes i casos d'ús, un llenguatge de programació orientat a objectes, en aquest projecte s'ha triat fer-lo amb java, utilitzant l'IDE Android Studio, els seus pertinents wireframes per tal de visualitzar el programa i un testing del programa.

El nom triat per a l'aplicació es "GiveltBack" i es un programa natiu per a android, la seva finalitat es compilar persones i objectes que deixis i enrecordar-te de quan les ha deixat, que i a qui.

NF1:Planificació

Estudi de mercat

Per tal de fer un estudi de mercat, primer s'ha de definir la nostra activitat la qual consisteix en **creació i desenvolupament d'aplicacions per a mòbil**.

Un cop sabem quin és la nostra activitat podem començar a definir qui son la nostra competencincia i demanda del sector.

Competencia:

Nom empresa	Oferta	Preu
NeoAttack	Posicionament web LinkBuilding Auditoria SEO Disseny Web Disseny web Botiga online Anàlítica web Google shopping Publicitat en adWords Community Managers Influencers Social Media Disseny Gràfic Branding inbounding Marketing	entre 1.000 - 1.000.000
Hello Media Group	Buisness services Branding & social media Total performance	entre 1.000-5.000€
Secuoyas	Dissenys personalitzats	entre 1.000-5.000€

Planificació

En la planificació del projecte, s'ha fet un diagrama de GANTT per poder veure en quin període de temps s'ha plantejat fer el projecte i tenir una aproximació de si la "death line" està ben posada o no s'ha plantejat alguns aspectes per el qual la data no està ben fixada.

En aquest cas està dividit en 4 seccions, els quatre nuclis formatius que el conformen.

En el primer apartat el Nucli Formatiu 1:Planificació, en el qual entren els apartats de:

- ☐ Estudi de mercat
- ☐ Planificació
- ☐ Pressupost del material i els recursos
- ☐ DAFO

El qual té una durada de 10h.

En el segon apartat el Nucli Formatiu 2: Disseny del projecte, en el que tenim els següents apartats:

- ☐ Anàlisi funcional
- ☐ Diagrama de casos d'ús
- ☐ Wireframes
- ☐ Diagrama del sistema de fitxers -classes- llibreries
- ☐ Model ER i Relacional de la base de dades

El qual té una durada de 16h.

El tercer apartat es el Nucli Formatiu 3:Implementació del projecte, conformat per:

- ☐ Programació
- ☐ Testing

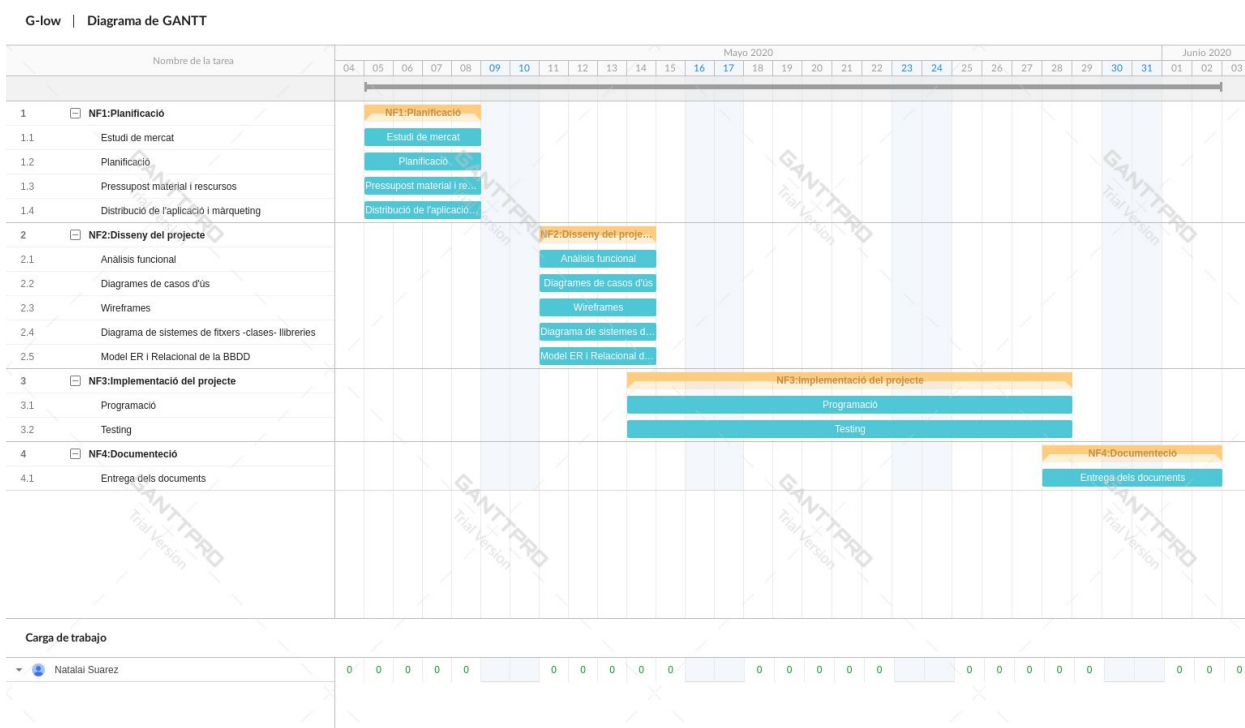
El que té més hores de totes amb un total de 57 hores.

L'últim apartat és el Nucli Formatiu 4: Documentació, en el qual entren tots els apartats que tenen a veure amb documentar i formatar:

- ☐ Documentació general
- ☐ Manual d'usuari
- ☐ Manual d'instal·lació

Amb 16 hores.

Fent un total de 99 hores més extensió del projecte.



Pressupost materials i recursos

El pressupost és un apartat molt important a l'hora de la creació d'un projecte, ja que és necessari per saber si tindrem despeses o podem cobrir tots els gastos que comporta el crear una aplicació o programa, en aquesta ocasió es individual, per compte propi, sense cap empresa en quant al tipus de despesa sobretot és a l'hora de desenvolupar el projecte, el preu que tindriem si cobresim un sou de 30€ l'hora que es lo que cobren aproximadament la majoria de "freelancers" es de 1.710€, però com que ho fem per compte propi i no ens cobrarem a nosaltres mateixos ens sortirà gratuït el desenvolupament del programa.

En quant a materials per portat a cap tot el projecte he utilitzat el meu portàtil amb un valor d'entre 900 - 1.000€ aquest ordinador es 2 en 1, es un portatil que es capaç de convertirse en tablet, a mes de tenir Touch pas, és a dir que tota la pantalla es tactil, amb un sistema operatiu windows 10 de x64, amb targeta de xarxa wifi i bluetooth 4.0, thinkpad de marca Lenovo, té una pantalla de 15.6 pulzades, un procesador Intel Core i5 de cinquena generació, una memoria RAM DDR3 de 8GB amb una tarjeta gràfica NVIDIA GeForce 840M.

El que no sortirà gratuït és la implementació del nostra programa a la Google Store o Play Store el qual té un cost de 20€

Els materials que utilitzarem en el desenvolupament del nostre projecte com a IDE i base per a programar serà Android Studio el cual es un programa de codi lliure i d'ús gratuït.

Distribució de l'aplicació i màrqueting

En quant a la distribució del nostre producte, que en el nostre cas, consisteix en una aplicació per a dispositius mòbils en concret una aplicació de codi natiu per a Android.

Per aquest motiu s'ha trobat que la millor manera de distribució de l'aplicació és mig llançant una tenda virtual, Play Store, però posar una aplicació en aquesta tenda virtual no és gratuït s'haurà de fer una inversió inicial de 20€ per tal de poder gaudir de descarregar el nostre programa amb un sol clic mitjançant la Play Store.



android



Google Play

DAFO

El tipo de análisis DAFO es una eina d'estudi d'una empresa o projecte per saber la situació en la que es troba, per tal d'analitzar les seves situacions externes (Amenaces i Oportunitats) i les seves característiques internes (Debilitats i fortaleeses) en una taula, per tal de veure amb més claredat quins són els nostres punts forts i quins són dolents, per tal de portar a cap la creació del nostre projecte o empresa.

Debilitats La nula visibilitat de comercialització	Amenaces Existeix molta competència en quant a creació de programes nadius per a Android
Fortaleeses Experiència amb diferents tipus de llenguatges de programació. Hi ha demanda. Preu de venda del producte	Oportunitats Implantació d'un nou estil d'app

NF2: Disseny del projecte

Analisi funcional

Anàlisi funcional:

En l'anàlisi funcional, es posaran tots els apartats que puguin necessaris per aclarir el funcionament de l'aplicació, de l'aparença de la mateixa i del backend.

Explicació del projecte:

Aquest projecte consisteix en una aplicació mòbil, feta amb androidStudio, la qual té com a objectiu principal, recordar-te dels objectes o diners que deixes a altres persones.

Objectius del projecte:

- Utilitzar la base de dades del telèfon, per tal que encara que tanquis l'aplicació es queden guardats els objectes que has deixat.
- Que tota l'aplicació es trobi en mínim en català i anglès.
- Tingui una interfaz simple però útil.
- Aplicació feta per a diverses mides de dispositius

Disseny de personatges i escenaris:

Aquest és l'apartat del disseny de tipus d'usuari que poden arribar a utilitzar la nostra aplicació(Personatges) i en quina situació es troben a l'hora d'utilitzar la (escenaris) i amb quina utilitat i finalitat la volen utilitzar.

Fer tant personatges com escenaris es molt util a l'hora de fer un projecte gran, ja que al definir el tipus de persona i com s'utilitzen ens facilitarà fer certes tasques necessàries a realitzar com els diagrames de casos d'ús o l'aparença de la mateixa aplicació

- Personatges:
 - Maria, 20 anys, acostumada a utilitzar aplicacions de xarxes socials com Instagram. Té uns coneixements bàsics sobre l'ús d'aplicacions mòbils, per tant no se li dificulta utilitzar noves apps si són visualment agradables i intuïtives.
 - Joan, 40 anys, és una persona molt ocupada que tampoc utilitza aplicacions que no siguin necessàries, no és molt hàbil amb la tecnologia.

- Escenaris:

Maria:

- Ella se n'adona que li ha deixat roba a una amiga, però mai s'recorda de demanar-li que si li pot demanar, per això va a l'aplicació i afegeix un nom a la roba, troba una foto de la mateixa i la posa per tal de visualitza que és el que ha deixat i indica quina a qui se l'ha deixat.

Joan:

- Ell és una persona que mai es recorda de tot el que deixa, per això un dia es troba amb aquesta aplicació i decideix posar els objectes que més deixa per tal de només hi ha d'indica qui ha deixat que.

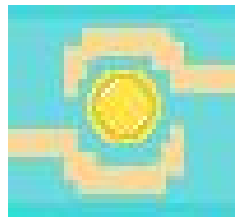


Nom, Logo i eslògan

Una cosa molt important per a una empresa o producte és el seu nom, el logo i l'eslògan.

El nom del programa creat per aquest projecte és "GiveltBack" que significa "Tornal de regres" el nom fa referència a l'aplicació, ja que es un programa per enrecordar les coses que has deixat a altres persones.

El logo com es pot veure té tres colors principals, blau, carn i groc. representa una mà retornant una cosa a l'altra.



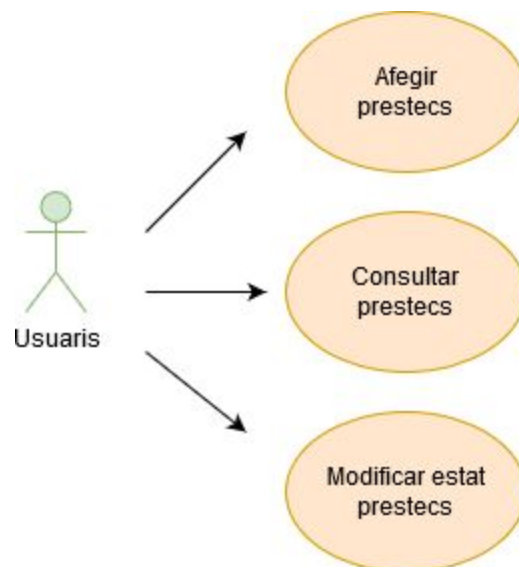
En quant a eslògan aquesta aplicació no es té cap, però es una bona idea el tenir-ne un si es una empresa o el nom no te correlació amb el tipus d'aplicació que vols vendre.

Els diagrames son una forma de saber l'organització interna i externa de l'aplicació, per tenir més claredat de com volem desenvolupar-la en quan comencen a picar codi, per això, a continuació estan els quatre diagrames, el diagrama de casos d'ús, el wireframes, els diagrames de sistemes de fitxers, el diagrama de entidad/relació i per últim el diagrama de relacions de base de dades.

Diagrama de casos d'ús

El diagrama de casos d'ús, es el diagrama que defineix quin és l'ús que li pot donar l'usuari a l'aplicació o programa.

Com es pot veure gràficament l'usuari pot fer principalment tres accions, pot afegir préstecs, pot consultar-los i pot modificar-los.

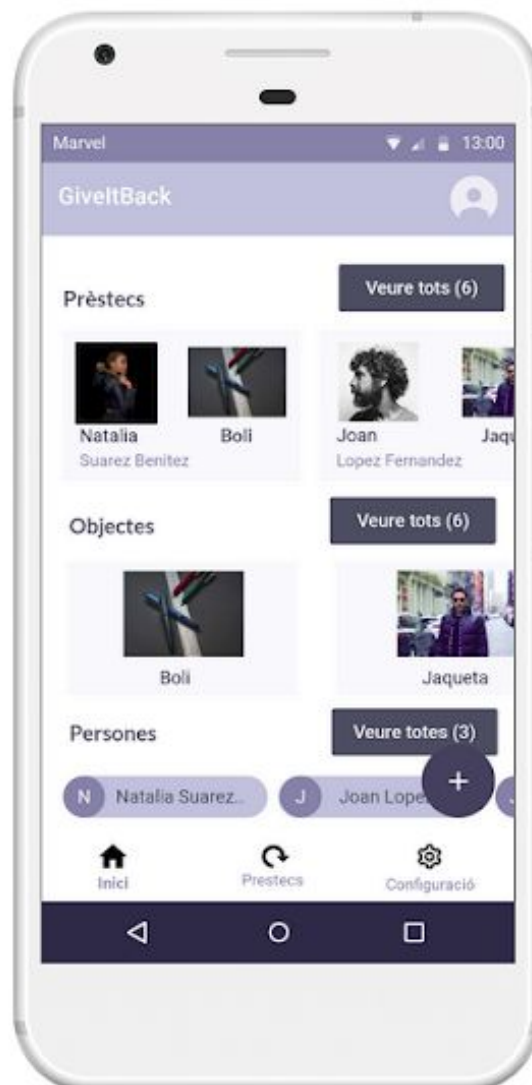


Wireframes

Només s'ha fet tres wireframes per tal de tenir una impressió de com ha de ser l'aparença de les pantalles del programa

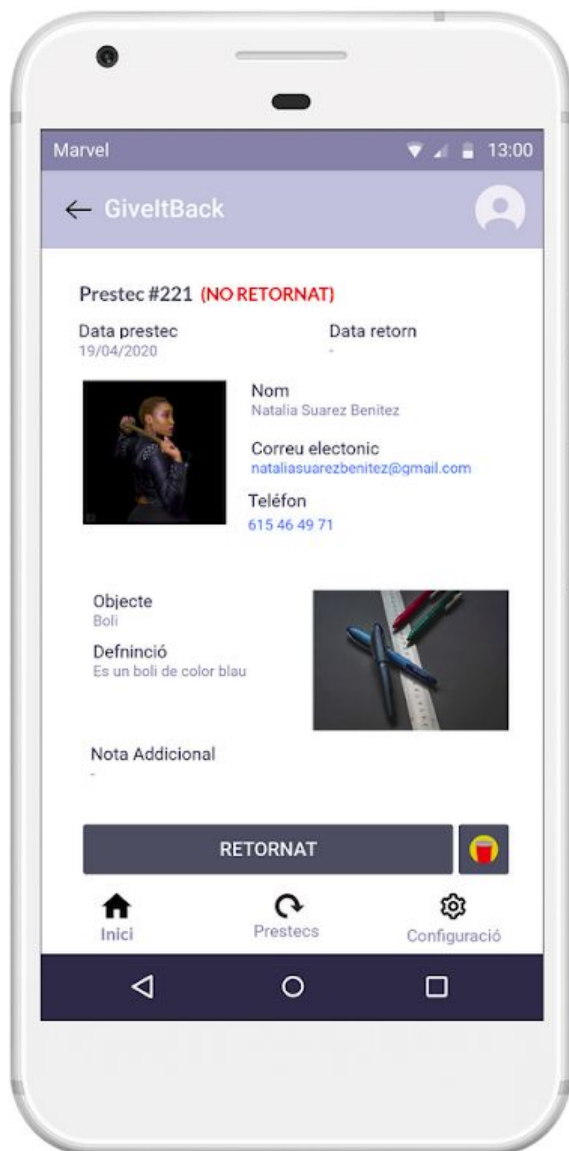
Wireframe 1:

El primer wireframe presentat és el següent, dalt de tot es troba els préstecs amb un botó a la dreta per mostrar-los tots, ja que el carrusel pot ser difícil de veure us amb claredat, El mateix passa amb els objectes i les persones implementades. Un cop cliques sobre d'algun preste,objecte o persona, s'obrirà una nova finestra amb tota la informació extesa.



Wireframe 2:

El segon wireframe mostra quan fas clic sobre préstecs, es la finestra amb la informació extesa, te tota la informació necessària a més de dos botons un per marcar que el préstec ha estat retornat i un segon botó per esborrar el préstec.



Wireframe 3:

El tercer wireframe es el que es veu quan volem afegir un préstec nou, els dropdowns contenen les persones o objectes que prèviament estiguin a la base de dades, també hi ha una opció d'afegir nou objecte o persones en cadascun dels dropdowns pertinents. amb un datePicker per triar quin dia vas deixar-lo prestat i una nota adicional.

The wireframe shows a mobile application interface for adding a new loan. The screen is titled "Nou préstec" (New loan) and is part of the "GiveltBack" app. The interface includes the following elements:

- Header:** A purple header bar with a back arrow, the text "GiveltBack", and a user profile icon.
- Title:** "Nou préstec" in bold black text.
- Form Fields:**
 - Persona:** A dropdown menu labeled "Dropdown lable" with a downward arrow.
 - Objecte:** A dropdown menu labeled "Dropdown lable" with a downward arrow.
 - Data de retorn:** A date picker field with the format "__ / __ / __".
 - Nota adicional:** A text area with placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam gravida venenatis accumsan. In mi massa, tempus". A character count "110/120" is displayed at the bottom right of the text area.
- Action Button:** A dark blue button labeled "AFEGIR" (Add) at the bottom of the form.
- Bottom Navigation Bar:** A dark blue bar with three icons and labels: a house icon for "Inici", a circular arrow icon for "Prestecs", and a gear icon for "Configuració".
- Android Bar:** The standard Android navigation bar at the very bottom.

Wireframe menú:

El menu té tres opcions Inici, Préstecs i Configuració.

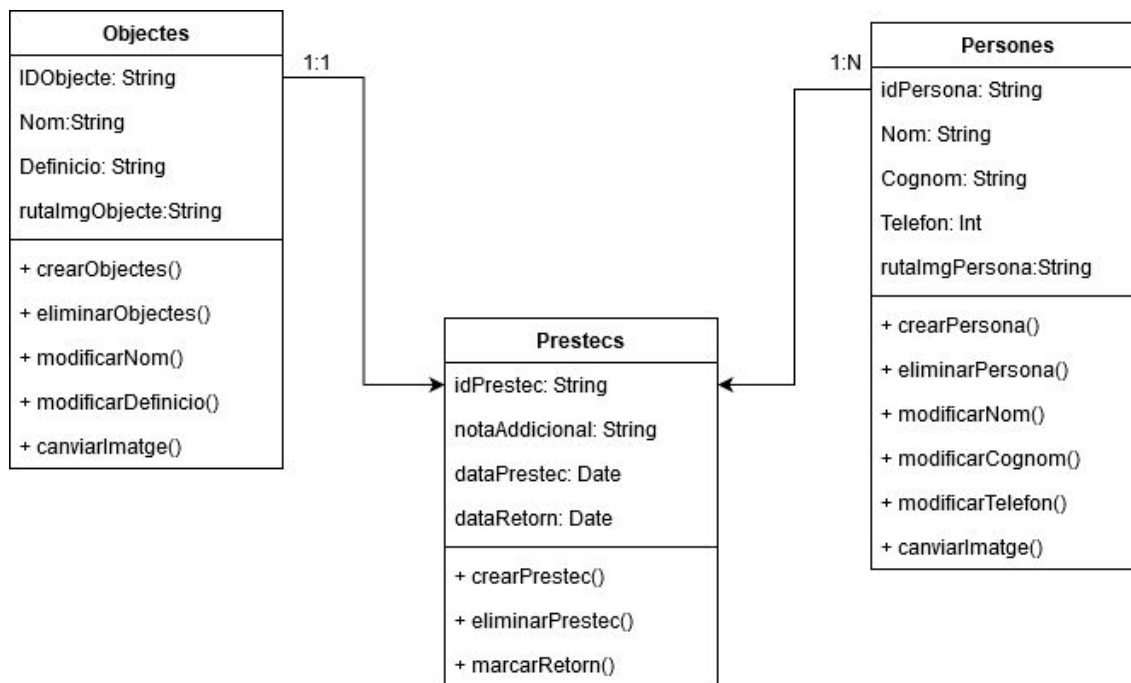


Diagrames de sistemes de fitxers

El diagrama de sistemes de fitxers es un diagrama on es presenten els diferents objectes que existiran en aquest programa, com es pot observar, tindrà tres objectes:

- ❑ préstecs
- ❑ Objectes
- ❑ persones

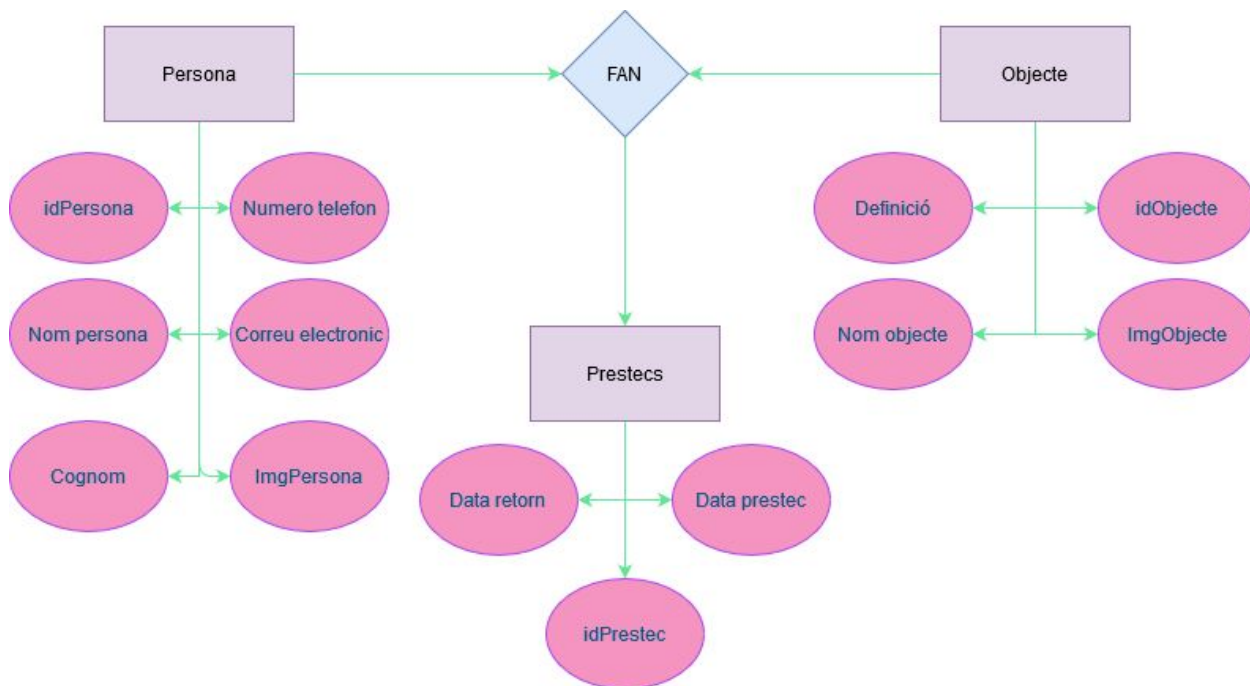
En cadascun d'ells es mostra els seus atributs i mètodes.



Model E/R i relacional de la base de dades

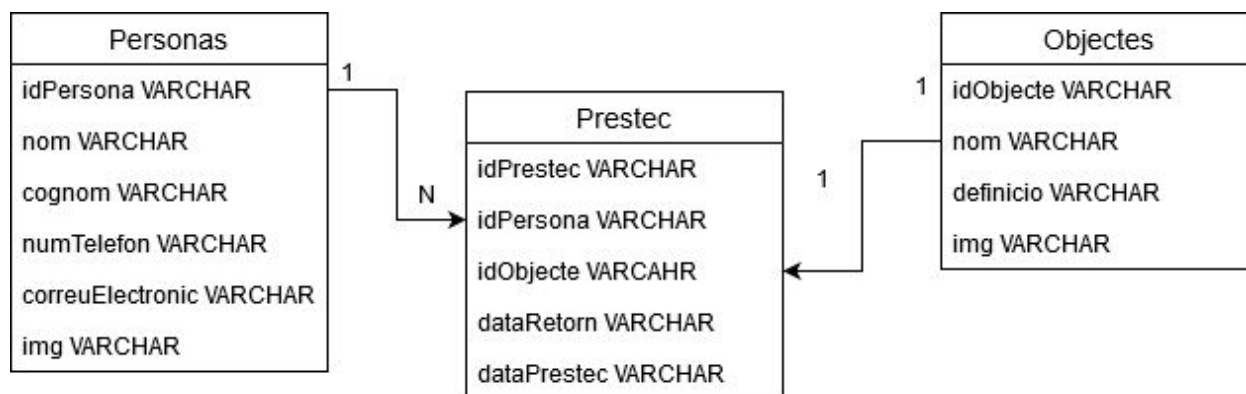
Model Entitat/Relació:

El diagrama entitat/relació mostra quina relació hi ha entre les entitats de l'aplicació i quina acció fan per tal de relacionar-se.



Model relacional de la base de dades:

Aquestes son les taules de la base de dades com es pot observar tenim tres taules, la relació de la taula persona amb la taula préstec es de 1 a molts, una persona pot tenir més d'un préstec, objectes es relaciona amb préstec 1 a 1, ja que un objecte només la pots deixar a una persona, no pots deixar el mateix objecte a dues persones a la vegada.



NF3:Implementació

Codi implementat

Codi mainActivity:

```
public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        //////////////////////////////////////

        // Predeterminat per la app

        //////////////////////////////////////

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        BottomNavigationView navView = findViewById(R.id.nav_view);

        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.

        AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
            R.id.navigation_home, R.id.navigation_dashboard, R.id.navigation_notifications)
            .build();

        NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);
        NavigationUI.setupWithNavController(navView, navController);
    }
}
```

```

////////////////////////////////////
// RecyclerViews + Observers de la db
////////////////////////////////////

AppDatabase app = DbClient.getClient(getApplicationContext());

final RecyclerView recyclerViewPretec = (RecyclerView)
findViewById(R.id.recycleViewPretec);

app.pretecDao().getAllCompleet().observe(this, new Observer<List<PretecCompleet>>() {

    @Override

    public void onChanged(List<PretecCompleet> pretecs) {

        LinearLayoutManager horizontalLayout = new LinearLayoutManager(MainActivity.this,
LinearLayoutManager.HORIZONTAL, false);

        recyclerViewPretec.setLayoutManager(horizontalLayout);

        recyclerViewPretec.setAdapter(new CardPretecAdapter(pretecs, MainActivity.this));

    }

});

final RecyclerView recyclerViewObjecte = (RecyclerView)
findViewById(R.id.recycleViewObject);

app.objecteDao().getAll().observe(this, new Observer<List<Objecte>>() {

    @Override

    public void onChanged(List<Objecte> objectes) {

        LinearLayoutManager horizontalLayout = new LinearLayoutManager(MainActivity.this,
LinearLayoutManager.HORIZONTAL, false);

        recyclerViewObjecte.setLayoutManager(horizontalLayout);

```

```

        recyclerViewObjecte.setAdapter(new CardObjectAdapter(objectes, MainActivity.this));
    }

});

final RecyclerView recyclerViewPersona = (RecyclerView)
findViewById(R.id.recycleViewPersones);

app.personaDao().getAll().observe(this, new Observer<List<Persona>>() {

    @Override

    public void onChanged(List<Persona> personas) {

        LinearLayoutManager horizontalLayout = new LinearLayoutManager(MainActivity.this,
LinearLayoutManager.HORIZONTAL, false);

        recyclerViewPersona.setLayoutManager(horizontalLayout);

        recyclerViewPersona.setAdapter(new CardPersonaAdapter(personas,
MainActivity.this));

    }

});

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Esdeveniments click
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

findViewById(R.id.fabOne).setOnClickListener(new View.OnClickListener() {

    @Override

    public void onClick(View view) {

        // Create new fragment and transaction

        /*  Fragment newFragment = new NewPrestecFragment();

```



```
FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();

// Replace whatever is in the fragment_container view with this fragment,
// and add the transaction to the back stack if needed
transaction.replace(R.id.nav_host_fragment, newFragment);
transaction.addToBackStack(null);

// Commit the transaction
transaction.commit();*/

Intent intent = new Intent(MainActivity.this, NewPrestecActivity.class);
startActivity(intent);

}

});
```



Codi fragment objecte:



codi viewObjecte

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout  
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    tools:context=".ui.notifications.NotificationsFragment">
```

```
    <LinearLayout
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="0dp"
```

```
        android:orientation="vertical">
```

```
        <TableLayout
```

```
            android:layout_width="match_parent"
```


```
            android:layout_height="wrap_content"
```

```
            android:layout_margin="16dp">
```

```
                <TableRow
```

```
                    android:layout_width="match_parent"
```

```
                    android:layout_height="111dp"
```



```
android:layout_marginTop="16dp">
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:layout_weight="1"
```

```
    android:orientation="vertical">
```

```
<TextView
```

```
    style="@style/TextAppearance.MaterialComponents.Overline"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="@string/objecte" />
```

```
<TextView
```

```
    android:id="@+id/txtObjecteNom"
```

```
    style="@style/TextAppearance.MaterialComponents.Body1"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Boli" />
```

```
<TextView
```

```
    style="@style/TextAppearance.MaterialComponents.Overline"
```

```
    android:layout_width="match_parent"
```



```
android:layout_height="wrap_content"

android:layout_marginTop="8dp"

android:text="@string/definicio" />
```

```
<TextView

android:id="@+id/txtObjecteDefinicio"

style="@style/TextAppearance.MaterialComponents.Body1"

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:text="Es un boli de color blau" />
```

```
</LinearLayout>
```

```
<ImageView

android:id="@+id/imgObjecte"

android:layout_width="100dp"

android:layout_height="132dp"

android:layout_marginLeft="16dp"


android:src="@mipmap/boli_object" />
```

```
</TableRow>
```

```
</TableLayout>
```

```
</LinearLayout>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```



codi activityObjecte

public class NewObjecteActivity extends Activity {

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState)
;

setContentView(R.layout.fragment_input_object);

// Submit de l'objecte



```
findViewById(R.id.buttonAfegirObjecte).setOnClickListener(new View.OnClickListener() {
```

```
@Override
```

```
public void onClick(View view) {
```

```
String nom = ((TextView) findViewById(R.id.txtObjecte)).getText().toString();
```

```
String def = ((TextView) findViewById(R.id.txtDefinicio)).getText().toString();
```

```
String uid = UUID.randomUUID().toString().replaceAll("-", "").toUpperCase();
```

```
Objecte obj = new Objecte();
```



```
obj.idObjecte = uid;
```

```
obj.nom = nom;
```

```
obj.definico = def;
```

```
obj.img = "";
```

```
DbClient.getClient(getApplicationContext()).objecteDao().insert(obj);
```

```
Intent resultIntent = new Intent();
```




```
resultIntent.putExtra("idObjecte", uid);
```

```
resultIntent.putExtra("nomObjecte", nom);
```

```
setResult(Activity.RESULT_OK, resultIntent);
```

```
finish();
```

```
}
```

```
));
```

```
}
```

```
public class NewObjecteActivity extends Activity {
```

```

@Override

public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.fragment_input_object);

    // Submit de l'objecte

    findViewById(R.id.buttonAfegirObjecte).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

    String nom = ((TextView) findViewById(R.id.txtObjecte)).getText().toString();

    String def = ((TextView) findViewById(R.id.txtDefinicio)).getText().toString();

    String uid = UUID.randomUUID().toString().replaceAll("-", "").toUpperCase();

    Objecte obj = new Objecte();

    obj.idObjecte = uid;

    obj.nom = nom;

    obj.definicio = def;

    obj.img = "";

    DbClient.getClient(getApplicationContext()).objecteDao().insert(obj);

    Intent resultIntent = new Intent();

    resultIntent.putExtra("idObjecte", uid);

```



```
resultIntent.putExtra("nomObjecte", nom);

setResult(Activity.RESULT_OK, resultIntent);
finish();
}

));
}
```



Conclusions

Aquest ha estat un credit de síntesis molt diferent de l'usual, per això m'he trobat amb moltes més dificultats a l'hora de fer-lo del que normalment em costaria, m'he trobat més desmotivada per el projecte del que mai creuria i amb la dificultat afegida de que es un credit de segon any de grau superior m'he trobat superada per les meves expectatives de voler fer-lo bé i no trobar la manera. Però tot i això aquest projecte m'ha servit molt, per trobar les meves mancances en quant a programació i les coses que puc millorar per superar els projectes que tingui més endavant.