

# CHEAPY

Robert Ripoll  
Daniel Re  
Sergi Boadas  
Luis Leon  
Quim Pascual  
Alex Segura

## **Idea de negoci**

L'idea darrera la nostra aplicació és permetre els usuaris una manera de contactar entre ells per tal de poder intercanviar productes a canvi d'una compensació econòmica. Els usuaris tenen la capacitat de penjar anuncis dels seus productes, veure els anuncis dels altres usuaris i contactar amb els propietaris dels productes que desitgin per tal d'arribar a un acord econòmic entre les parts.

Cal remarcar que la nostra aplicació no conté mètodes de pagament ni transaccions online, sinó que actua més aviat com un pont per tal de comunicar venedors i compradors.

La nostra manera de monetitzar l'aplicació vindria a ser a través d'anuncis que terceres parts volguessin mostrar a l'aplicació, a més de permetre ressaltar els anuncis de productes en venda per tal de que es prioritzin davant d'altres productes. Una altra opció seria mantenir el model de negoci actual, actuar com a pont entre venedor i comprador i, en cas que l'aplicació tingui èxit, podríem anar afegint transaccions online si la nostra base d'usuaris així ho desitja.

Hem tingut en ment que hi ha moltes aplicacions semblants al mercat, així que hauríem de diferenciar-nos de la resta d'alguna manera per tal d'acabar tenint un bon lloc al mercat.

## **Requeriments funcionals i no funcionals definitius (implementats)**

**Requeriments funcionals:** el client ha de poder registrar-se o connectar-se a la seva conta, veure la llista de productes i fer clic al producte que li interressi. En mode invitat sempre es podran veure els anuncis de productes en venda, i accedir als detalls d'aquests, però mai fer transaccions, ni contactar amb altres usuaris via missatgeria instantània. En cas que hagi vist el producte i vulgui contactar amb l'usuari que el ven, ha de poder enviar-li missatges. La cerca de productes en venda es pot filtrar per: categoria, venedor, ubicació, preu, preu negociable, intercanvi acceptat. L'usuari ha de poder veure les seves dades del perfil així com modificar-les en cas que ho desitgi, també pot accedir als seus productes venuts i comprats fins al moment i clicar-los per obtenir més informació.

**Requeriments no funcionals:** bon rendiment de l'aplicació, accessos ràpids a la base de dades així com assegurar la seguretat de les dades dels usuaris.

## **Documentació de l'API REST. Podeu fer servir la del Apiary**

## Resource Group

### IMATGE

**POST**`/imatges`

Pujar una/múltiples imatge(s)

**Example URI****POST** `http://polls.apibluprint.org//imatges`**Request**[Show](#)**Response** `200`

### IMATGE EN CONCRET

**GET**`/imatges/{nomArxiu}`

Veure una imatge

**Example URI****GET** `http://polls.apibluprint.org//imatges/nomArxiu`**URI Parameters**[Hide](#)**Response** `200`**DELETE**`/imatges/{nomArxiu}`

Esborrar una imatge

**Example URI****DELETE** `http://polls.apibluprint.org//imatges/nomArxiu`**URI Parameters**[Hide](#)**Response** `200`

### CATEGORIES

**GET**`/categories`

Llistar Categories

**Example URI****GET** `http://polls.apibluprint.org//categories`**Response** `200`[Show](#)

### PRODUCTES

**GET**`/productes`

Llistar Productes (en Venda)

**Example URI****GET** `http://polls.apibluprint.org//productes`**Request** `Llistar productes en venda`[Show](#)**Response** `200`[Show](#)**Request** `Llistar productes de categories en concret i amb preu no negociable, ordenades ascendentment per preu i nom`[Show](#)

Response 200

Show

POST

/productes

Crear Producte

#### Example URI

**POST** http://polls.apibluprint.org//productes

#### Request

Show

Response 200

Show

### PRODUCTE

GET

/productes/{producte\_id}

Veure Producte

#### Example URI

**GET** http://polls.apibluprint.org//productes/producte\_id

#### URI Parameters

Hide

**Request** Veure un Producte venut o comprat per l'usuari identificat

Show

Response 200

Show

**Request** Veure un Producte NO propi

Show

Response 200

Show

PUT

/productes/{producte\_id}

Actualitzar Producte

#### Example URI

**PUT** http://polls.apibluprint.org//productes/producte\_id

#### URI Parameters

Hide

**Request** Incrementar visites

Show

Response 200

Show

**Request** Actualitzar nom i categoria

Show

Response 200

Show

DELETE

/productes/{producte\_id}

Eliminar Producte

#### Example URI

**DELETE** http://polls.apibluprint.org//productes/producte\_id

#### URI Parameters

Hide

Response 200

### TRANSACCIÓ D'UN PRODUCTE

POST

/productes/transaccio

Vendre Producte

#### Example URI

**POST** http://polls.apibluprint.org//productes/transaccio

#### Request

Show

Response 200

Show

DELETE

/productes/transaccio

Anul·lar venda

Example URI

DELETE http://polls.apibluprint.org//productes/transaccio

Response 200

Show

## VALORACIÓ DEL COMPRADOR DE LA TRANSACCIÓ D'UN PRODUCTE

POST

/productes/transaccio/valoracio

Valorar transacció

Example URI

POST http://polls.apibluprint.org//productes/transaccio/valoracio

Request

Show

Response 200

Show

DELETE

/productes/transaccio/valoracio

Esborrar valoració feta

Example URI

DELETE http://polls.apibluprint.org//productes/transaccio/valoracio

Response 200

Show

## USUARIS

POST

/usuaris

Crear Usuari

Example URI

POST http://polls.apibluprint.org//usuaris

Request

Show

Response 200

Show

## PRODUCTES PREFERITS D'UN USUARI

GET

/usuaris/jo/favorites

Veure favorites

Example URI

GET http://polls.apibluprint.org//usuaris/jo/favorites

Response 200

Show

## PRODUCTE PREFERIT D'UN USUARI

POST

/usuaris/jo/favorites/

Afegir Producte

Example URI

POST http://polls.apibluprint.org//usuaris/jo/favorites/

Response 200

DELETE

/usuaris/jo/favorites/

Treure Producte

Example URI

**DELETE** <http://polls.apibluprint.org//usuaris/jo/favorites/>

Response 200

## AUTENTICAR USUARI

**POST**

</usuaris/autenticar>

Autenticar

### Example URI

**POST** <http://polls.apibluprint.org//usuaris/autenticar>

### Request

Show

Response 200

Show

## DESAUTENTICAR USUARI

**POST**

</usuaris/desautenticar>

Desautenticar

### Example URI

**POST** <http://polls.apibluprint.org//usuaris/desautenticar>

Response 200

## USUARI

**GET**

[/usuaris/{usuari\\_id}](/usuaris/{usuari_id})

Veure Usuari

### Example URI

**GET** [http://polls.apibluprint.org//usuaris/usuari\\_id](http://polls.apibluprint.org//usuaris/usuari_id)

### URI Parameters

Hide

Response 200

Show

Request Es vol veure el perfil d'un usuari diferent al de la sessió iniciada

Show

Response 200

Show

## USUARI AUTENTICAT

**PUT**

</usuaris/jo>

Actualitzar Usuari

### Example URI

**PUT** <http://polls.apibluprint.org//usuaris/jo>

### Request

Show

Response 200

Show

**DELETE**

</usuaris/jo>

Esborrar Usuari

### Example URI

**DELETE** <http://polls.apibluprint.org//usuaris/jo>

Response 200

## VALORACIONS D'UN USUARI ESPECÍFIC

**GET**

[/usuaris/{usuari\\_id}/valoracions](/usuaris/{usuari_id}/valoracions)

Veure valoracions de l'usuari

#### Example URI

**GET** [http://polls.apibluprint.org//usuaris/usuari\\_id/valoracions](http://polls.apibluprint.org//usuaris/usuari_id/valoracions)

#### URI Parameters

[Hide](#)

Response **200**

[Show](#)

### PRODUCTES VENDA D'UN USUARI

**GET**

[/usuaris/{usuari\\_id}/vendes](#)

Llistar Productes venuts

#### Example URI

**GET** [http://polls.apibluprint.org//usuaris/usuari\\_id/vendes](http://polls.apibluprint.org//usuaris/usuari_id/vendes)

#### URI Parameters

[Hide](#)

Response **200**

[Show](#)

### PRODUCTES COMPRATS D'UN USUARI

**GET**

[/usuaris/{usuari\\_id}/compres](#)

Llistar Productes comprats

#### Example URI

**GET** [http://polls.apibluprint.org//usuaris/usuari\\_id/compres](http://polls.apibluprint.org//usuaris/usuari_id/compres)

#### URI Parameters

[Hide](#)

Response **200**

[Show](#)

### CONVERSACIONS

**GET**

[/conversacions](#)

Llistar Conversacions

#### Example URI

**GET** <http://polls.apibluprint.org//conversacions>

Response **200**

[Show](#)

**POST**

[/conversacions](#)

Crear Conversació

#### Example URI

**POST** <http://polls.apibluprint.org//conversacions>

#### Request

[Show](#)

Response **200**

[Show](#)

### CONVERSACIÓ

**PUT**

[/conversacions/{conversa\\_id}](#)

Marcar Missatges de la Conversació com a llegits

#### Example URI

**PUT** [http://polls.apibluprint.org//conversacions/conversa\\_id](http://polls.apibluprint.org//conversacions/conversa_id)

#### URI Parameters

[Hide](#)

Response **200**

[Show](#)

**DELETE**

[/conversacions/{conversa\\_id}](#)

Esborrar Conversació



Example

**DELETE** [http://polls.apibluprint.org/conversacions/versa\\_id](http://polls.apibluprint.org/conversacions/versa_id)

URI Parameters

Hide

Response

## MISSATGES D'UNA CONVERSA

**GET**

[/conversacions/{conversa\\_id}/missatges](#)

Llistar Missatges de la Conversació

Example

**GET** [http://polls.apibluprint.org/conversacions/versa\\_id/missatges](http://polls.apibluprint.org/conversacions/versa_id/missatges)

URI Parameters

Hide

Response

Show

**POST**

[/conversacions/{conversa\\_id}/missatges](#)

Enviar Missatge

Example

**POST** [http://polls.apibluprint.org/conversacions/versa\\_id/missatges](http://polls.apibluprint.org/conversacions/versa_id/missatges)

URI Parameters

Hide

Request

Show

Response

Show

**DELETE**

[/conversacions/{conversa\\_id}/missatges](#)

Esborrar un Missatge

Example

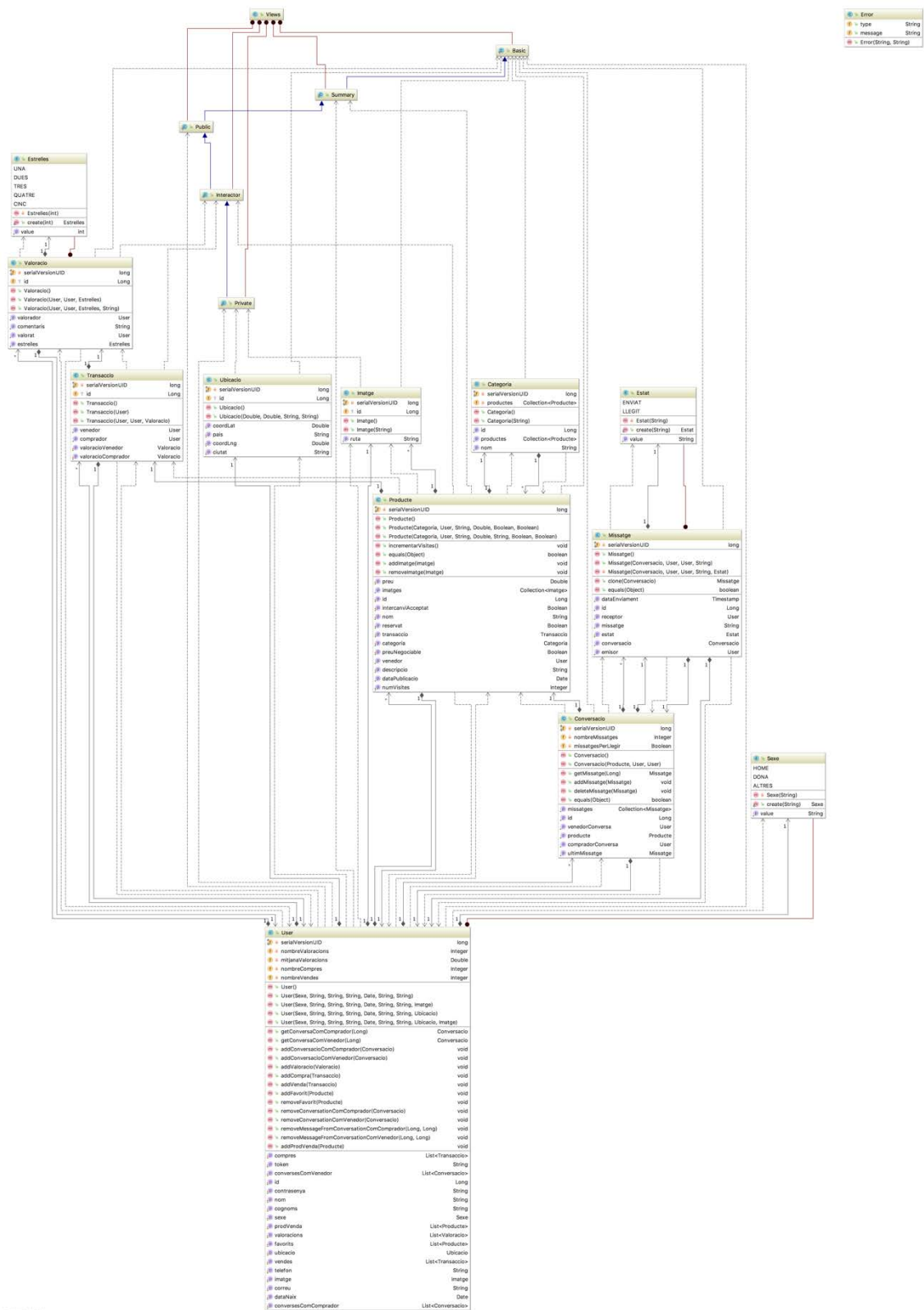
**DELETE** [http://polls.apibluprint.org/conversacions/versa\\_id/missatges](http://polls.apibluprint.org/conversacions/versa_id/missatges)

URI Parameters

Hide

Response

### Diagrama entitat relació de la base de dades del servidor



## **URLs dels repositoris git, i de l'aplicació a OpenShift si l'heu pujat**

Les URLs dels repositoris del git són les següents:

- La part del client → [https://github.com/UngarMax/pds\\_client](https://github.com/UngarMax/pds_client)
- La part del servidor → [https://github.com/UngarMax/pds\\_servidor](https://github.com/UngarMax/pds_servidor)

## **Extres que heu fet servir respecte a l'aplicació d'exemple (p.e. Retrofit, Picasso, Websockets, Firebase, ...)**

Un extra que hem fet servir és el Firebase que és una plataforma de desenvolupament en el núvol de google, aquesta plataforma és una eina molt potent per crear aplicacions web i per mòbil, per tant firebase el que fa és crear una API per guardar i sincronitzar dades en el núvol en temps real, ja sigui per hosting, test lab, emmagatzematge etc...

En el nostre cas el que hem fet és utilitzar un client de firebase per tal de poder enviar notificacions al dispositiu el qual li enviem un missatge, per tant cada cop que un usuari envia un missatge a un altre, utilitzem firebase per tal d'enviar una notificació a aquest usuari a partir del seu token. En el server s'ha creat una crida corresponent per assignar el token a un usuari a partir del client.

L'extra que hem utilitzar per a l'emmagatzematge de les imatges ha sigut Minio, una API que ens permet emmagatzemar arxius en un servidor diferent al que es troba el servidor REST del backend. Aquesta API ens ha sigut útil per tal de guardar externament, en un altre servidor, les imatges de perfil dels usuaris i les imatges dels productes anunciats.

Un altre extra utilitzat, és el Picasso, el qual permet poder carregar les nostres imatges a partir d'una URL, i així poder-les mostrar correctament a les imageView de la nostre app.

## **Instruccions d'us**

Al iniciar l'aplicació apareix una splashscreen amb el logo de l'app a on depenen de si ja estas, o no, loggejat al servidor, et redirigirà a la pantalla de login o a la pantalla principal dins la teva conta que estava oberta.

Un cop introduïdes les dades i connectat a una conta, l'usuari pot visualitzar tota la llista de productes des del menú fent clic a l'opció "productes" i contactar al venedor a través de la missatgeria(aquesta segona funció només serà possible en cas que l'usuari estigui connectat). Des d'aquesta pantalla també és possible veure la política de privacitat de l'empresa fent clic a la franja inferior de la pantalla on diu "Privacy Policy".

Si l'usuari està connectat a la seva conta, podrà veure altres funcionalitats de l'aplicació fent clic al menú. Des d'aquest menú l'usuari podrà registrar un producte propi i emplenar les dades en relació a la venda d'aquest producte.

Des del menú l'usuari pot accedir al seu perfil, des del perfil pot editar les seves dades a través del botó "Editar Perfil" just a sota de la imatge. Si el que vol és veure els seus productes comprats i els seus productes venuts, ha de fer clic al carro de la compra que hi ha a sota de "Productes

Comprats” i “Productes a la Venda”, per més informació només ha de fer clic a sobre del producte desitjat.

També es pot visualitzar l’ubicació del moment de crear la compte en el maps (sempre hi quan el gps estigui obert) i també es poden visualitzar chats de la conversa. Aquests chats estaran lligats amb les converses existents i així al clicar a la conversa, et redirigirà a la conversa sencera desitjada.

Si el que es vol és modificar un producte que es té a la venda (perquè no es ven, perquè vols modificar el preu, etc.) el que has de fer és anar al perfil i a “Productes a la Venda”, aquí selecciona el producte en qüestió i després ha de fer clic al botó amb forma de llapis.

## **Problemes e incidències**

Un dels principals problemes que hem tingut els de la part d’Android és que quasi tot ens venia de nou i hem hagut de recolzar-nos molt de l’ajuda d’Internet. És el primer cop que fèiem una aplicació i això també s’ha reflectit a l’hora de recollir els requeriments de com volíem que fos la nostra aplicació i quines funcionalitats havia de ser capaç de desenvolupar.

El que volem dir amb això és que al principi ens pensàvem que algunes coses portarien menys feina del que realment portaven i això ens va obligar a anar acabant de modelar l’aplicació sobre la marxa. Per exemple, al principi havíem pensat que podríem implementar algunes funcionalitats que semblaven no portar gaire feina i, al posar-nos-hi, ens hem donat conta que la funcionalitat l’havíem de desglossar més i repartir més la implementació d’aquesta funcionalitat.

També hem tingut problemes amb el tema de l’apiary ja que s’anava modificant i això enrederia poder avançar feina nova i haver de fer els suposats refactors de les entitats de nou.

Els de backend ens hem trobat amb el repte del sistema de filtratge de productes. Aquest ha sigut especialment difícil amb el filtratge per distància. MariaDB no ve amb una funció, de fàbrica, que calculi la distància entre dues coordenades (latitud i longitud), i per tant, hem hagut de crear una funció SQL a l’import.sql. Vam tenir considerables problemes a l’hora de que aquesta funció SQL s’importés des d’aquest arxiu, ja que quan fèiem el CREATE FUNCTION des de Java ens donava error sintàctic, però des de la consola sí ens ho acceptava. Un cop adaptada la funció perquè s’admetés des de Java fent ús de l’import.sql, feia falta crear el sistema de filtratge, i conseqüentment contemplar totes les possibilitats de filtratge (fer-lo modular). Fer-lo modular implica que tant es podien aplicar tots els filtres a la vegada o bé només un únic filtre. També va implicar considerar l’ús de varis operadors relacionals a la vegada, i per tant considerar, per exemple, un rang de preu.

Un altre problema que hem tingut ha sigut a l’hora de valorar, ja que no hi ha un mètode objectiu. Hem puntuat segons el treball que ens ha portat la tasca i el valor que afegia a l’aplicació però tot i així ens ha semblat una part dura ja que mai saps exactament quina és la puntuació real que mereix aquella tasca.