

# TOYMAKER

Proyecto final DAW 2017  
IES Campanillas

Moisés Jiménez Martín

## Contenido

<b>1. MANUAL DE USUARIO.....</b>	<b>2</b>
1.1 Descripción de la aplicación .....	2
1.2 Funcionalidades y características.....	2
1.3 Uso y requisitos .....	2
<b>2. MANUAL TÉCNICO .....</b>	<b>3</b>
2.1 Modelo de datos .....	3
2.2 Lenguajes utilizados y frameworks.....	3
2.3 Servidor .....	4
<b>3. ANEXOS.....</b>	<b>5</b>
Anexo 1 .....	5
Anexo 2 .....	6
Anexo 3.....	7
Anexo 4.....	8
Anexo 5.....	9
Anexo 6 .....	10
Anexo 7 .....	11

# 1. MANUAL DE USUARIO

## 1.1 Descripción de la aplicación<sup>1</sup>

ToyMaker es una app móvil de venta de diseños 3D que asocia estos a una cuenta de Astroprint, otra aplicación web y móvil desde la que puedes imprimirlos. De igual forma, estos diseños adquiridos quedan registrados en la colección propia del usuario.

A través de la aplicación puedes crearte una cuenta de Astroprint o acceder con una ya creada anteriormente.

Además de la app móvil, ToyMaker cuenta con un backend web que permite al diseñador (sólo uno en este caso) subir los diseños, su imagen, descripción, designarle un precio etc. También se pueden controlar las colecciones a las que pertenecen estos diseños.

## 1.2 Funcionalidades y características

- APP:
  - Inicio de sesión
  - Creación de cuenta de usuario
  - Añadir diseños a tu colección de Astroprint
  - Añadir colección de diseños a tu colección de Astroprint
  - Logout
- Backend
  - Login
  - Gestión de diseños
  - Gestión de colecciones (categorías)
  - Control de usuarios y diseños adquiridos.

## 1.3 Uso y requisitos

Para el uso de la app se necesita tener una cuenta de Astroprint, o crearte una. El uso de la app es bastante intuitivo, clicando en los distintos elementos puedes ver y comprar los diseños de tu interés mediante tu cuenta de GooglePlay o I-tunes (proceso de compra aún no implementado).

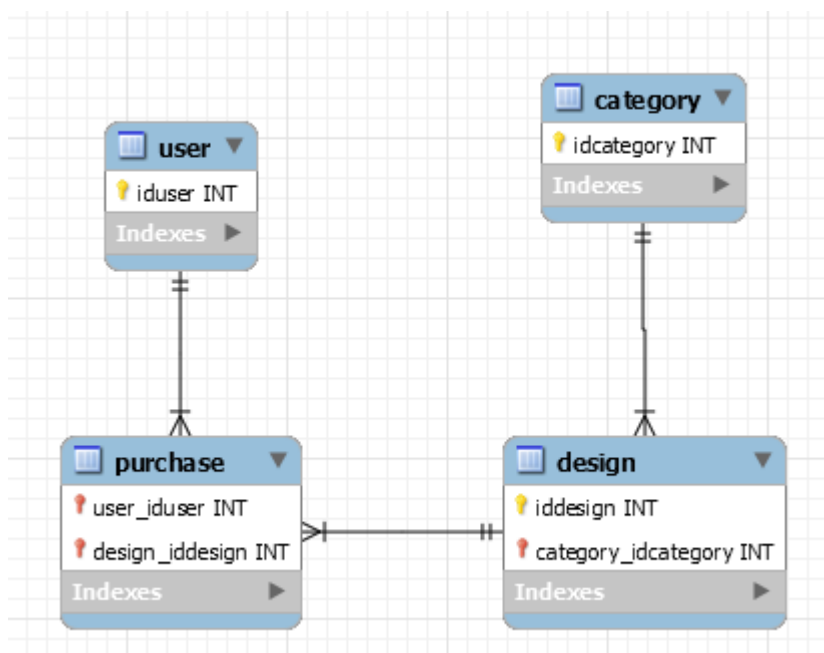
---

<sup>1</sup> Capturas en Anexos 1-3

## 2. MANUAL TÉCNICO

### 2.1 Modelo de datos

El modelo de datos es el siguiente:



La base de datos se ha desarrollado en PostgreSQL

### 2.2 Lenguajes utilizados y frameworks<sup>2</sup>

Para la realización de la App Móvil se ha utilizado IONIC 2 Framework. Por tanto se ha seguido la estructura de un proyecto de Angular 2. Esta app se comunica a través de llamadas HTTP con una API desarrollada en la misma web en la que se encuentra el backend.

Para el login, la creación de usuarios, y añadir colecciones o diseños a la cuenta del usuario se utiliza la API de Astroprint, así como su sistema de tokens.

Para las imágenes y los archivos de diseños 3D se usa el servidor Amazon S3, por lo que la carga y descarga de los mismos se realiza a través de llamadas a la API de Amazon.

Para la realización del panel de gestión y la API se ha utilizado el Framework MVC Phalcon.

<sup>2</sup> Capturas en Anexos 4-7

El diseño se ha realizado en CSS3 utilizando SASS

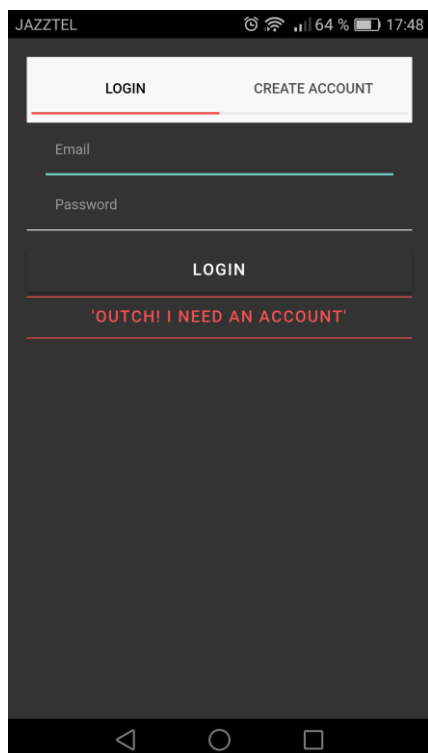
## 2.3 Servidor

El backend está alojado en un servidor compartido con una instalación Debian y la web está corriendo en NginX. Sólo permite el acceso a través de SSH.

## 3. ANEXOS

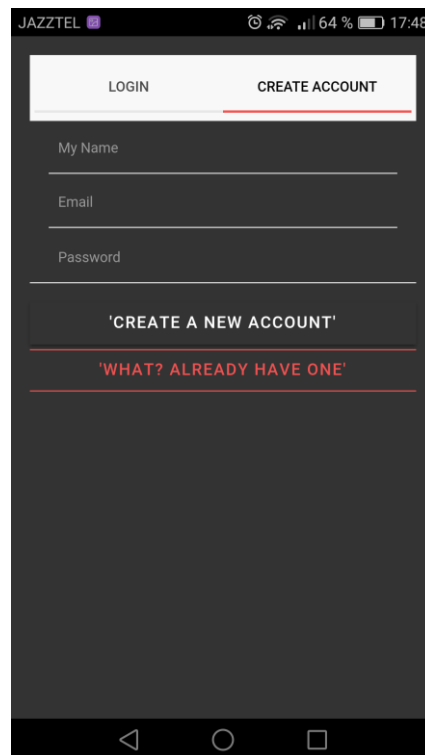
### Anexo 1

Captura de Login



The screenshot shows a mobile application interface for login. At the top, there is a header with 'JAZZTEL' on the left and status icons (signal, Wi-Fi, battery at 64%, and time 17:48) on the right. Below the header is a white bar with two tabs: 'LOGIN' (selected) and 'CREATE ACCOUNT'. Under the 'LOGIN' tab, there are two input fields: 'Email' and 'Password'. Below these fields is a 'LOGIN' button. At the bottom, there is a red text message: 'OUCH! I NEED AN ACCOUNT'.

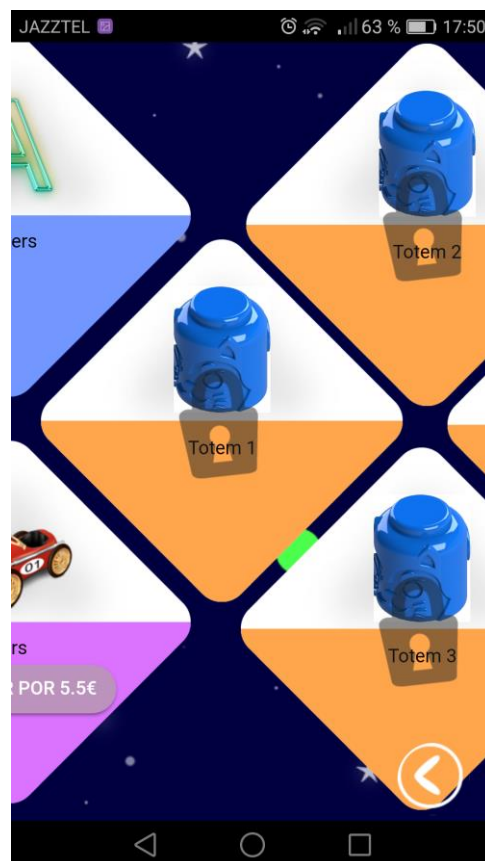
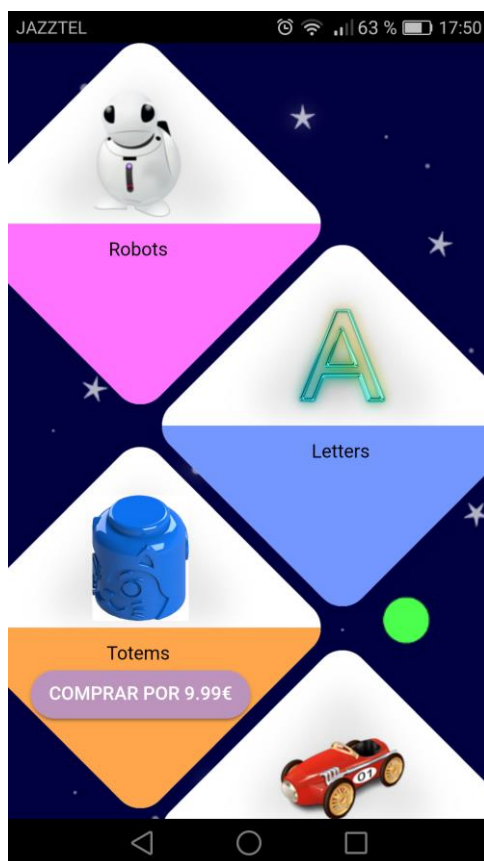
Captura de Create Account



The screenshot shows a mobile application interface for creating a new account. At the top, there is a header with 'JAZZTEL' on the left and status icons (signal, Wi-Fi, battery at 64%, and time 17:48) on the right. Below the header is a white bar with two tabs: 'LOGIN' and 'CREATE ACCOUNT' (selected). Under the 'CREATE ACCOUNT' tab, there are three input fields: 'My Name', 'Email', and 'Password'. Below these fields is a 'CREATE A NEW ACCOUNT' button. At the bottom, there is a red text message: 'WHAT? ALREADY HAVE ONE'.

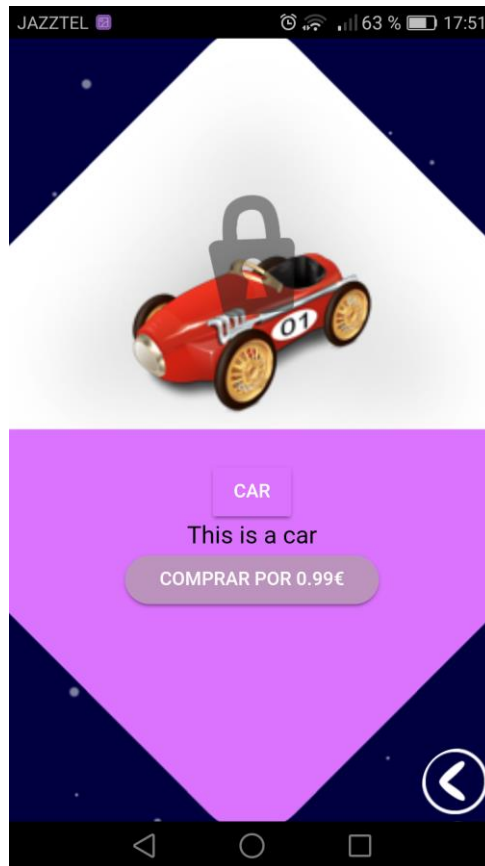
## Anexo 2

Pantalla principal y desplegable horizontal



## Anexo 3

Vista de detalle de la compra





## Anexo 4

Primeras comprobaciones IONIC

```

/////////////////SUBSCRIBE EVENTS/////////////////
//Logout event
events.subscribe('logOut', () => {
  this.setHome(LoginPage);
});

//Login Event
events.subscribe('logIn', () => {
  this.setHome(HomePage);
  console.log("hola");
  console.log("2");
});
this.auth.initialiceToken().then( //Descomentar para login
  () => {
    if(this.auth.refreshToken){
      this.setHome(HomePage);
    }else{
      this.setHome(LoginPage)
    }
  });
platform.ready().then(() => {
  // Okay, so the platform is ready and our plugins are available.
  // Here you can do any higher level native things you might need.
  StatusBar.styleDefault();
  SplashScreen.hide();
});
}

setHome(page){
  this.auth.getAccountDetails().then((data)=>{
    this.translate.setDefaultLang(data['language']);
  },(error)=>{
    console.log('error recuperando el Lenguaje');
  });
  if(page == HomePage){
    console.log(this.user.getFirstTime());
    this.user.getFirstTime().then(()=>{
      if(this.user.firstTime.showWelcome){
        this.nav.setRoot(HomePage, {showWelcome: this.user.firstTime.showWelcome});
      }else{
        this.nav.setRoot(page);
      }
    });
  }else{
    this.nav.setRoot(page);
  }
}
}

```

## Anexo 5

HTML Pantala principal

```

8  <ion-content (on-scroll)="scrolled()">
9  <!--<ion-scroll scrollX="true" scrollY="true" zoom="false">-->
10  <div class="columna" *ngFor="let categoria of categorias; let i = index;">
11  <button #elemento id="{{categoria.nombre}}" ion-button class="menu" (click)="(i ==
12  categorias.length-1) ? logOut() : despliega(categoria.nombre, i)" >
13  <p>{{(i != categorias.length-1) ? ('HELLO' | translate) : ('LOGOUT' | translate) }}</p>
14  </button>
15  <ng-container *ngIf="isVisible == categoria.nombre" >
16  <div class="desplegable" id="desplegable" [ngStyle]="{'width.vw': ( i > 0 ?
17  categoria.subcategorias.length*27 : categoria.subcategorias.length*37) }">
18  <ul>
19  <li *ngFor="let subcategoria of categoria.subcategorias">
20  <button ion-button class="subMenu">
21  <p>{{subcategoria.nombre}}</p>
22  </button>
23  </li>
24  </ul>
25  </div>
26  </ng-container>
27  </div>
28  <span color='default' (click)="hideMe()" *ngIf="scrollButton" class="menu scrollButton">
29  </span>
30  <!--</ion-scroll -->
31  <div class="welcomeSlider" *ngIf="showWelcome">
32  <div class="slide-1" [ngStyle]="{'width.px': deviceDisplayWidth}">
33  <div class="texto">
34  <p>
35  {{ "ASTROTEXT" | translate }}
36  </p>
37  </div>
38  <div id="astro">
39  <span color='default' (click)="nextSlide()" class="nextSlideButton left">
40  </span>
41  </div>
42  </div>
43
44
45  <div class="slide-2" [ngStyle]="{'width.px': deviceDisplayWidth, 'margin-left.px':
46  deviceDisplayWidth}">
47  <div class="texto">
48  <p>
49  {{ "PRINTERTEXT" | translate }}
50  </p>
51  </div>
52  <div id="impre">
53  <button ion-button color="light" (click)="skipWelcome()" id="buttonWelcome">
54  Ok!
55  </button>
56  </div>
57  </div>
58  </ion-content>

```

## Anexo 6

Uso de SASS en Pantalla Principal

```
1 page-home{
2   button.menu, .desplegable button {
3     width: $buttonSize;
4     height: $buttonSize !important;
5     position: relative;
6     padding: 0;
7     overflow: hidden;
8     //background-color: blue;
9     -ms-transform: rotate(45deg); /* IE 9 */
10    -webkit-transform: rotate(45deg); /* Chrome, Safari, Opera */
11    transform: rotate(45deg);
12    border-radius: 10px;
13  }
14  span.button-inner{
15    overflow: hidden;
16    position: relative;
17    border-radius: inherit;
18    z-index: 1;
19  }
20  span.button-inner p{
21    z-index: 3;
22  }
23  span.button-inner::after{
24    content: ' ';
25    position: absolute;
26    width: $buttonSize + $buttonSize/2.8;
27    height: $buttonSize + $buttonSize/2.8 + 1;
28    margin-top: $buttonSize/2-1;
29    margin-left: $buttonSize/2-1;
30    -ms-transform: rotate(45deg); /* IE 9 */
31    -webkit-transform: rotate(45deg); /* Chrome, Safari, Opera */
32    transform: rotate(45deg);
33    z-index: 2;
34  }
35  .columnna:nth-child(5n+1) button .button-inner{
36    background-color: map-get($colors, first);
37  }
38  .columnna:nth-child(5n+2) button .button-inner{
39    background-color: map-get($colors, second);
40  }
41  .columnna:nth-child(5n+3) button .button-inner{
42    background-color: map-get($colors, third);
43  }
44  .columnna:nth-child(5n+4) button .button-inner{
45    background-color: map-get($colors, fourth);
46  }
47  .columnna:nth-child(5n+5) button .button-inner{
48    background-color: map-get($colors, fifth);
49  }
50  .columnna:nth-child(5n+1) button .button-inner::after{
51    background-color: map-get($colors, firstDark);
52  }
```

## Anexo 7

Muestra de acceso a API en Phalcon

```

369
370 /////////////////////////////////////////////////// DESIGNS ///////////////////////////////////
371 // Retrieves all designs
372 $app->get(
373     "/api/designs",
374     function () use ($app) {
375         $phql = "SELECT * FROM ToyMaker\\Admin\\Design ORDER BY idDesign";
376
377         $designs = $app->modelsManager->executeQuery($phql);
378
379         $data = [];
380
381         foreach ($designs as $design) {
382             $data[] = [
383                 "idDesign" => $design->idDesign,
384                 "imgUrl" => $design->imgUrl,
385                 "title" => $design->title,
386                 "description" => $design->description,
387                 "designUrl" => $design->designUrl,
388                 "available" => $design->available,
389                 "created_at" => $design->created_at,
390                 //"category" => $design->category
391             ];
392         }
393         $response = new Response();
394         $response->setJsonContent($data);
395         return $response;
396         //echo json_encode($data);
397     }
398 );
399
400 // Adds a new design
401 $app->post(
402     "/api/design",
403     function () use ($app) {
404
405         $design = $app->request->getJsonRawBody();
406
407         $phql = "INSERT INTO ToyMaker\\Admin\\Design ( imgUrl, title, description, designUrl,
408             available, category, created_at ) VALUES (:imgUrl:, :title:, :description:,
409             :designUrl
410             :, :available:, :category:, :created_at:)";
411
412         $status = $app->modelsManager->executeQuery(
413             $phql,
414             [
415                 "imgUrl" => $design->imgUrl,
416                 "title" => $design->title,
417                 "description" => $design->description,
418                 "designUrl" => $design->designUrl,
419                 "available" => $design->available,
420                 "created_at" => date("Y-m-d H:i:s"),
421                 "category" => $design->category
422             ]

```