

```
>ionic start crud blank
```

```
Integrate your new app with Capacitor to target native iOS and Android? No
```

```
Create free Ionic account? No
```

```
>cd crud
```

```
ionic g service services/storage
```

```
ionic cordova plugin add cordova-sqlite-storage
```

```
Are you sure you want to continue? Yes
```

home.page.html X

home.page.scss

TS home.page.ts

TS app.component.ts

TS storage.service.spec.ts

TS app.mo

src > app > home > home.page.html > ion-content > ion-button

```
1 <ion-header>
2   <ion-toolbar color="primary">
3     <ion-title>
4       Mercado Gente Feliz
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8
9 <ion-content>
10  <ion-item>
11    <ion-label position="floating">Nome:</ion-label>
12    <ion-input [(ngModel)]="newItem.nome"></ion-input>
13  </ion-item>
14
15  <ion-item>
16    <ion-label position="floating">Marca:</ion-label>
17    <ion-input [(ngModel)]="newItem.marca"></ion-input>
18  </ion-item>
19
20  <ion-item>
21    <ion-label position="floating">Preço:</ion-label>
22    <ion-input [(ngModel)]="newItem.preco"></ion-input>
23  </ion-item>
24  <br>
25
26
27  <ion-button color="secondary" expand="full" (click)="addItem()">Cadastrar Produto</ion-button>
28  <br>
29  <ion-toolbar color="primary">
30    <ion-title>
31      Produtos Cadastrados
32    </ion-title>
33  </ion-toolbar>
34  <br>
35  <ion-list #mylist>
```

<> home.page.html X

home.page.scss

TS home.page.ts

TS app.component.ts

TS storage.service.spec.ts

TS app.module.ts

src > app > home > <> home.page.html > ion-content > ion-button

```
36
37     <ion-item-sliding *ngFor="let item of items">
38
39         <ion-item>
40             <ion-label text-wrap>
41                 <h3>{{ item.nome }}</h3>
42                 <ion-text color="secondary">
43                     <p>{{ item.marca }}</p>
44                     <p>{{ item.preco }}</p>
45
46                 </ion-text>
47                 <p>{{ item.modified | date:'short' }}</p>
48             </ion-label>
49         </ion-item>
50
51         <ion-item-options side="end">
52             <ion-item-option color="secondary" (click)="updateItem(item)">Atualizar</ion-item-option>
53             <ion-item-option color="danger" (click)="deleteItem(item)">Deletar</ion-item-option>
54         </ion-item-options>
55     </ion-item-sliding>
56
57 </ion-list>
58
59 </ion-content>
60
```

src > app > home > TS home.page.ts > HomePage

```
1 import { Component, ViewChild } from '@angular/core';
2 import { StorageService, Item } from '../services/storage.service';
3 import { Platform, ToastController, IonList, NavController } from '@ionic/angular';
4
5
6 @Component({
7   selector: 'app-home',
8   templateUrl: 'home.page.html',
9   styleUrls: ['home.page.scss'],
10 })
11 export class HomePage {
12
13   items: Item[] = [];
14
15   newItem: Item = <Item>{};
16
17   @ViewChild('mylist')mylist: IonList;
18
19   constructor(private storageService: StorageService,
20     private plt: Platform,
21     private toastController: ToastController,
22     private nav: NavController) {
23     this.plt.ready().then(() => {
24       this.loadItems();
25     });
26   }
27
28   // CREATE
29   addItem() {
30     this.newItem.modified = Date.now(); // atribui a data atual para a propriedade do
31     this.newItem.id = Date.now(); // atribui a data atual para a propriedade do
32
33     this.storageService.addItem(this.newItem).then(item => { // adiciona um novo item
34       this.newItem = <Item>{};
35       this.showToast('Produto adicionado!');
36       this.loadItems(); // Atualiza os itens adicionados
37     });
38   }
39
40 }
```

```

39
40
41 // UPDATE
42 updateItem(item: Item) {
43     // a propriedade ID, não é atualizada pegamos do item da tela
44     // atualizamos as outras propriedades
45     item.nome = `${this.newItem.nome}`; // adiciona os dados da propriedade (dados
46     item.marca = `${this.newItem.marca}`;
47     item.preco = `${this.newItem.preco}`;
48     item.modified = Date.now();
49
50     this.storageService.updateItem(item).then(item => {
51         this.showToast('Produto Adicionado!');
52         this.mylist.closeSlidingItems(); // fecha o sliding Items depois de atual.
53         this.loadItems(); // Atualiza os itens atualizados
54     });
55 }
56
57 // DELETE
58 deleteItem(item: Item) {
59     this.storageService.deleteItem(item.id).then(item => {
60         this.showToast('Produto Excluido!');
61         this.mylist.closeSlidingItems(); // fecha o sliding Items depois de deletar
62         this.loadItems(); // Atualiza os itens atualizados
63     });
64 }
65
66 // Helper
67 async showToast(msg) { // exibe uma mensagem de 2 segundos
68     const toast = await this.toastController.create({
69         message: msg,
70         duration: 2000
71     });
72     toast.present();
73 }
74
75 loadItems() {
76     this.storageService.getItems().then(items => { // lê os itens do banco de dados e carrega n.
77         this.items = items;
78     });
79 }
80

```

home.page.html TS home.page.ts TS storage.service.ts X TS app.component.ts TS storage.service.spec.ts TS app.module.ts

src > app > services > TS storage.service.ts > StorageService

```
1 import { Injectable } from '@angular/core';
2 import { Storage } from '@ionic/storage';
3
4 export interface Item { // interface com os atributos definidos
5   id: number,
6   nome: string,
7   marca: string,
8   preco: string,
9   modified: number
10 }
11
12 const ITEMS_KEY = 'my-items'; // constante
13
14 @Injectable({
15   providedIn: 'root'
16 })
17 export class StorageService {
18
19   constructor(private storage: Storage) { }
20
21   // CREATE
22   addItem(item: Item): Promise<any> { //abstração para representar fluxos de
23     return this.storage.get(ITEMS_KEY).then((items: Item[]) => { //Obter os dados associado à chave fornecida
24       if (items) {
25         items.push(item); //adiciona o item no array de items
26         return this.storage.set(ITEMS_KEY, items); //seta que a chave key ITEMS_KEY, recebe o array de items
27       } else {
28         return this.storage.set(ITEMS_KEY, [item]);
29       }
30     });
31   }
32
33   // READ
34   getItems(): Promise<Item[]> { .....
35     return this.storage.get(ITEMS_KEY); ..... //Obter os dados associado à chave fornecida
36   }
37 }
```

<> home.page.html TS home.page.ts TS storage.service.ts X TS app.component.ts TS storage.service.spec.ts TS app.module

src > app > services > TS storage.service.ts > StorageService

```
38 // UPDATE
39 updateItem(item: Item): Promise<any> {
40   return this.storage.get(ITEMS_KEY).then((items: Item[]) => { // Obter os dados associado à chave fornecida
41     if (!items || items.length === 0) { // se não existir itens ou items = 0
42       return null;
43     }
44
45     let newItems: Item[] = []; // cria um array new Items
46
47     for (let i of items) { // percorre os itens armazenados
48       if (i.id === item.id) { // se o item da tela teve o mesmo id, a
49         newItems.push(item);
50       } else {
51         newItems.push(i);
52       }
53     }
54
55     return this.storage.set(ITEMS_KEY, newItems); // atualiza a lista
56   });
57 }
58
59 // DELETE
60 deleteItem(id: number): Promise<Item> {
61   return this.storage.get(ITEMS_KEY).then((items: Item[]) => {
62     if (!items || items.length === 0) {
63       return null;
64     }
65
66     let toKeep: Item[] = [];
67
68     for (let i of items) {
69       if (i.id !== id) {
70         toKeep.push(i);
71       }
72     }
73     return this.storage.set(ITEMS_KEY, toKeep);
74   });
75 }
76 }
```


src > app > TS app.module.ts > ...

```
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { RouteReuseStrategy } from '@angular/router';
4
5 import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
6 import { SplashScreen } from '@ionic-native/splash-screen/ngx';
7 import { StatusBar } from '@ionic-native/status-bar/ngx';
8
9 import { AppComponent } from './app.component';
10 import { AppRoutingModuleModule } from './app-routing.module';
11 import { IonicStorageModule } from '@ionic/storage';
12
13 @NgModule({
14   declarations: [AppComponent],
15   entryComponents: [],
16   imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModuleModule, IonicStorageModule.forRoot({
17     name: '__mydb',
18   driverOrder: ['indexeddb', 'sqlite', 'websql']
19   })],
20   providers: [
21     StatusBar,
22     SplashScreen,
23     { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
24   ],
25   bootstrap: [AppComponent]
26 })
27 export class AppModule {}
28
```


