Persistencia Poliglota

CouchDB
relax

pouchdb

TTROJO

Pannunzio, Faustino
Piñeiro, Eugenia
Sagüés, Ignacio

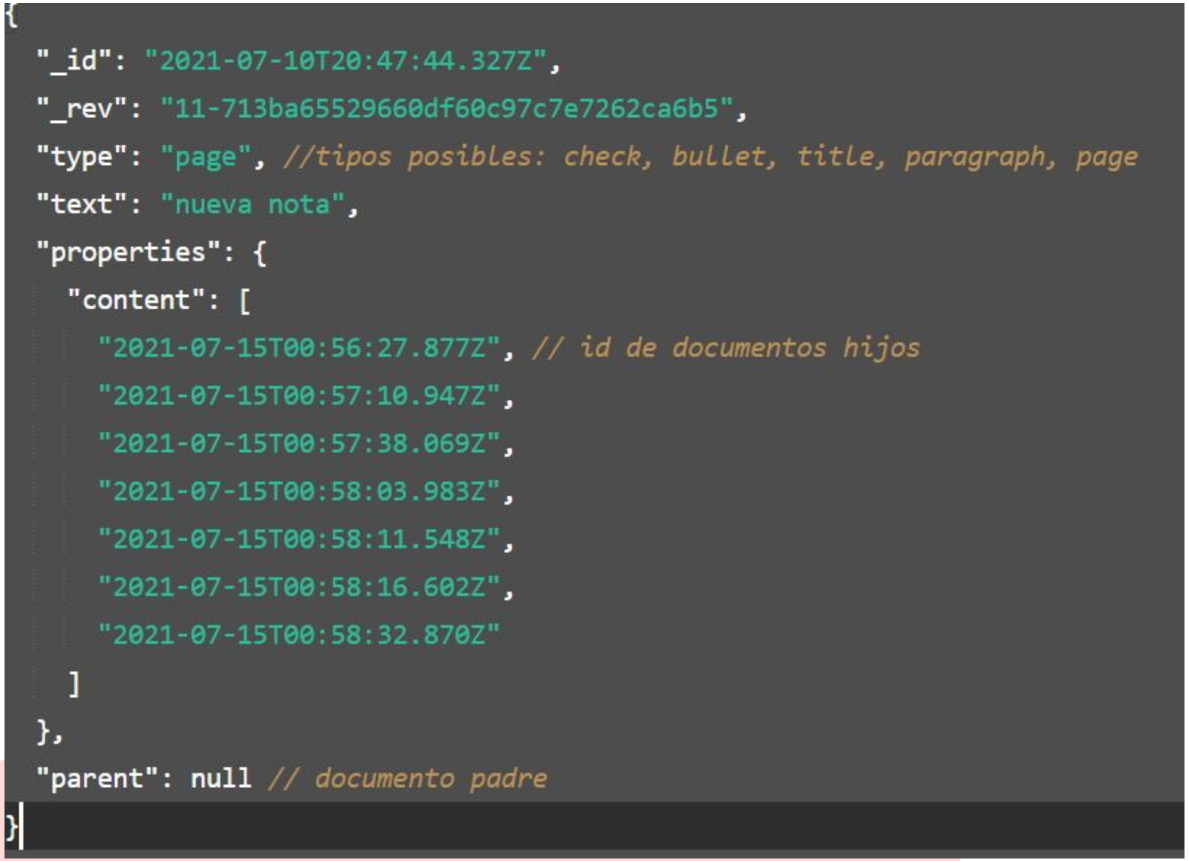
TABLA DE CONTENIDOS

01

SISTEMA DE NOTAS

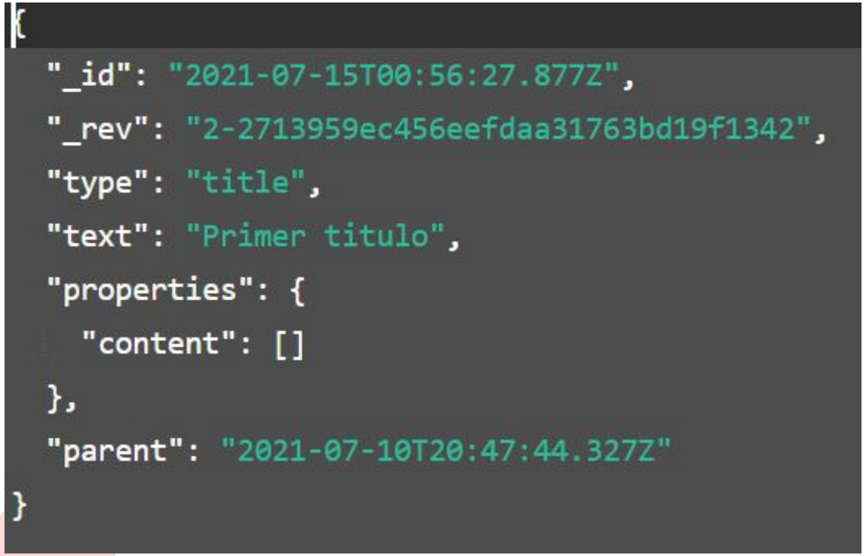
Introducción a la idea implementada

ESTRUCTURA DE LOS DOCUMENTOS



```
{
  "_id": "2021-07-10T20:47:44.327Z",
  "_rev": "11-713ba65529660df60c97c7e7262ca6b5",
  "type": "page", //tipos posibles: check, bullet, title, paragraph, page
  "text": "nueva nota",
  "properties": {
    "content": [
      "2021-07-15T00:56:27.877Z", // id de documentos hijos
      "2021-07-15T00:57:10.947Z",
      "2021-07-15T00:57:38.069Z",
      "2021-07-15T00:58:03.983Z",
      "2021-07-15T00:58:11.548Z",
      "2021-07-15T00:58:16.602Z",
      "2021-07-15T00:58:32.870Z"
    ]
  },
  "parent": null // documento padre
}
```

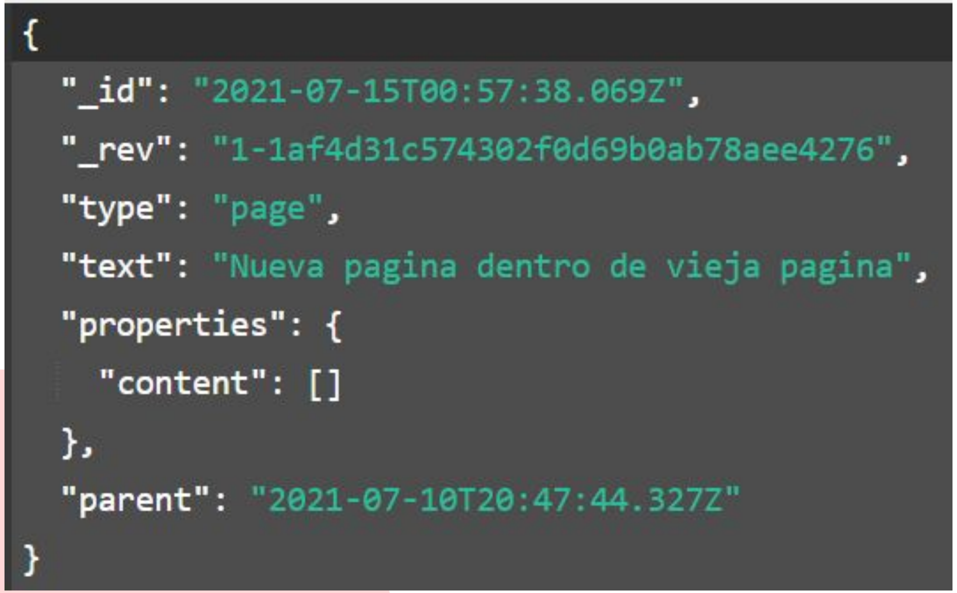
ESTRUCTURA DE LOS DOCUMENTOS



```
{
  "_id": "2021-07-15T00:56:27.877Z",
  "_rev": "2-2713959ec456eefdaa31763bd19f1342",
  "type": "title",
  "text": "Primer titulo",
  "properties": {
    "content": []
  },
  "parent": "2021-07-10T20:47:44.327Z"
}
```

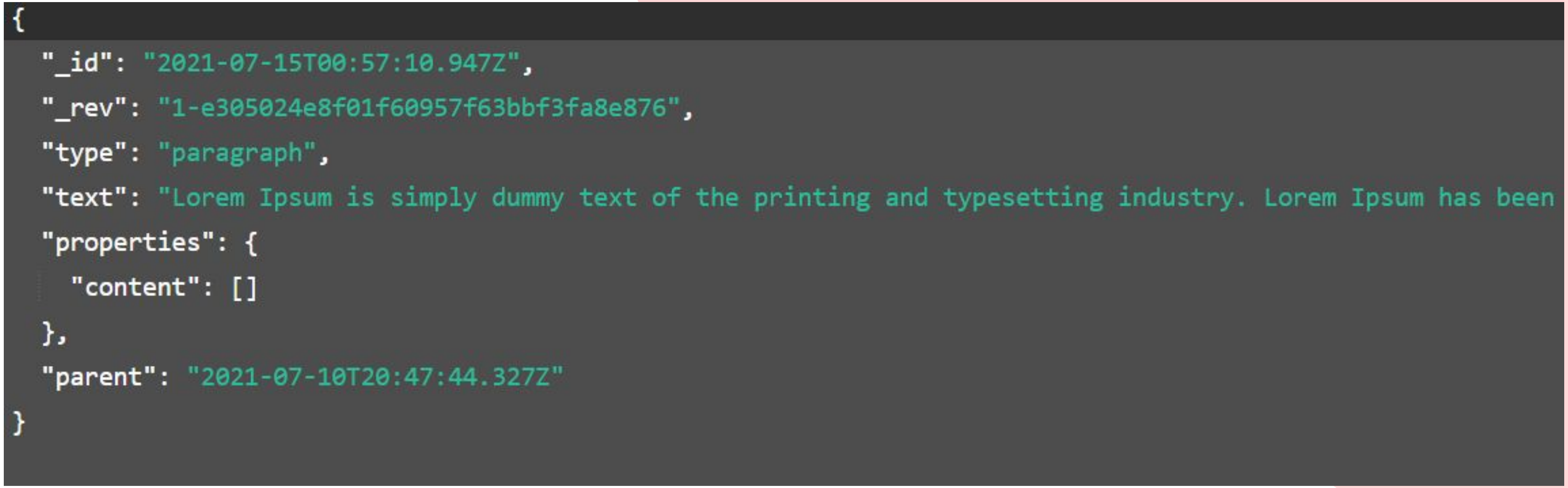
Primer titulo

Nueva pagina dentro de vieja pagina



```
{
  "_id": "2021-07-15T00:57:38.069Z",
  "_rev": "1-1af4d31c574302f0d69b0ab78aee4276",
  "type": "page",
  "text": "Nueva pagina dentro de vieja pagina",
  "properties": {
    "content": []
  },
  "parent": "2021-07-10T20:47:44.327Z"
}
```

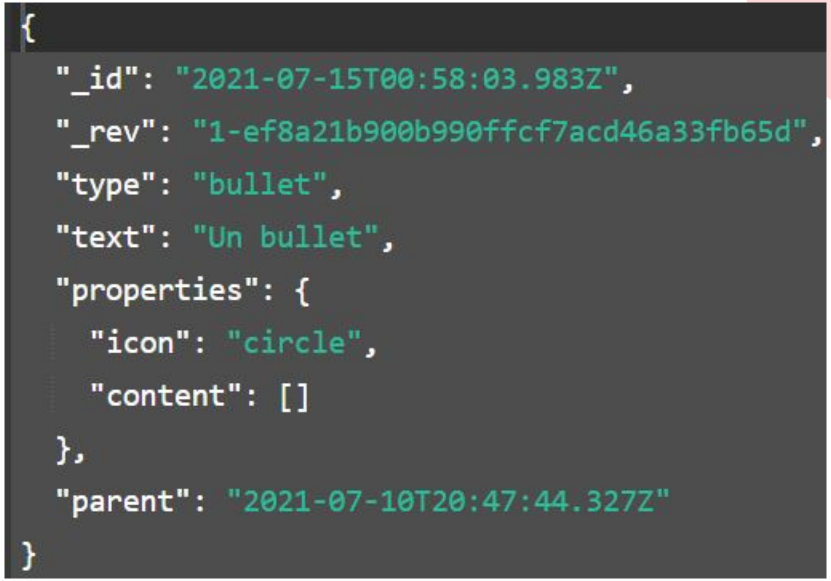
ESTRUCTURA DE LOS DOCUMENTOS



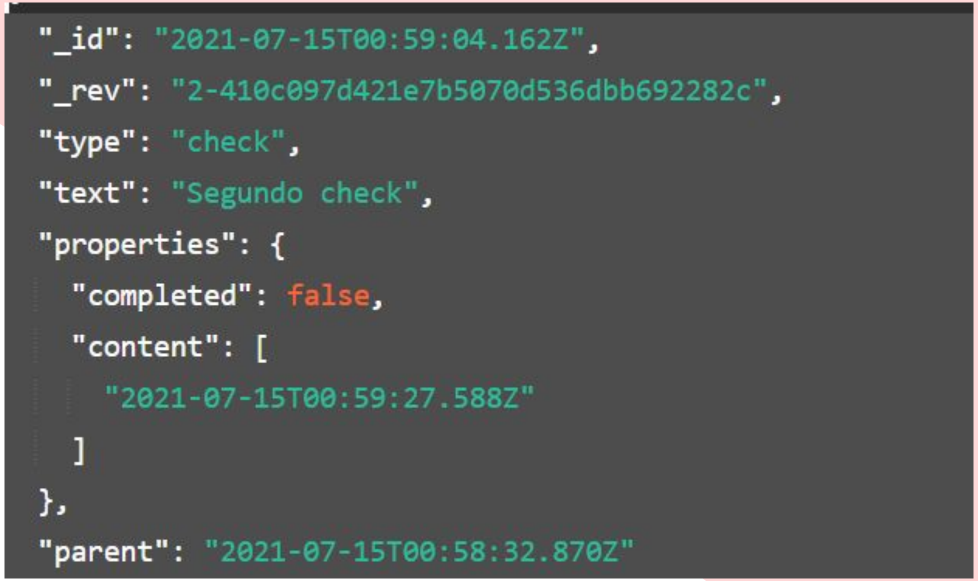
```
{
  "_id": "2021-07-15T00:57:10.947Z",
  "_rev": "1-e305024e8f01f60957f63bbf3fa8e876",
  "type": "paragraph",
  "text": "Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been
  "properties": {
    "content": []
  },
  "parent": "2021-07-10T20:47:44.327Z"
}
```

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

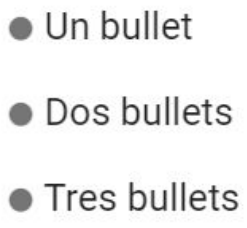
ESTRUCTURA DE LOS DOCUMENTOS



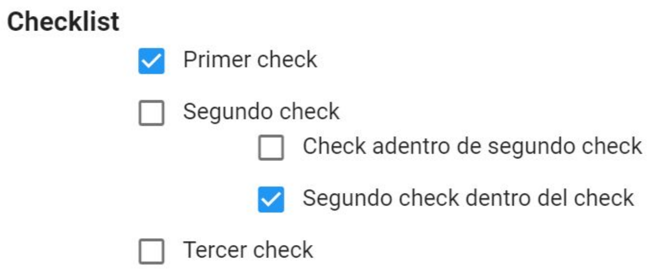
```
{
  "_id": "2021-07-15T00:58:03.983Z",
  "_rev": "1-ef8a21b900b990ffcf7acd46a33fb65d",
  "type": "bullet",
  "text": "Un bullet",
  "properties": {
    "icon": "circle",
    "content": []
  },
  "parent": "2021-07-10T20:47:44.327Z"
}
```



```
"_id": "2021-07-15T00:59:04.162Z",
"_rev": "2-410c097d421e7b5070d536dbb692282c",
"type": "check",
"text": "Segundo check",
"properties": {
  "completed": false,
  "content": [
    "2021-07-15T00:59:27.588Z"
  ]
},
"parent": "2021-07-15T00:58:32.870Z"
```



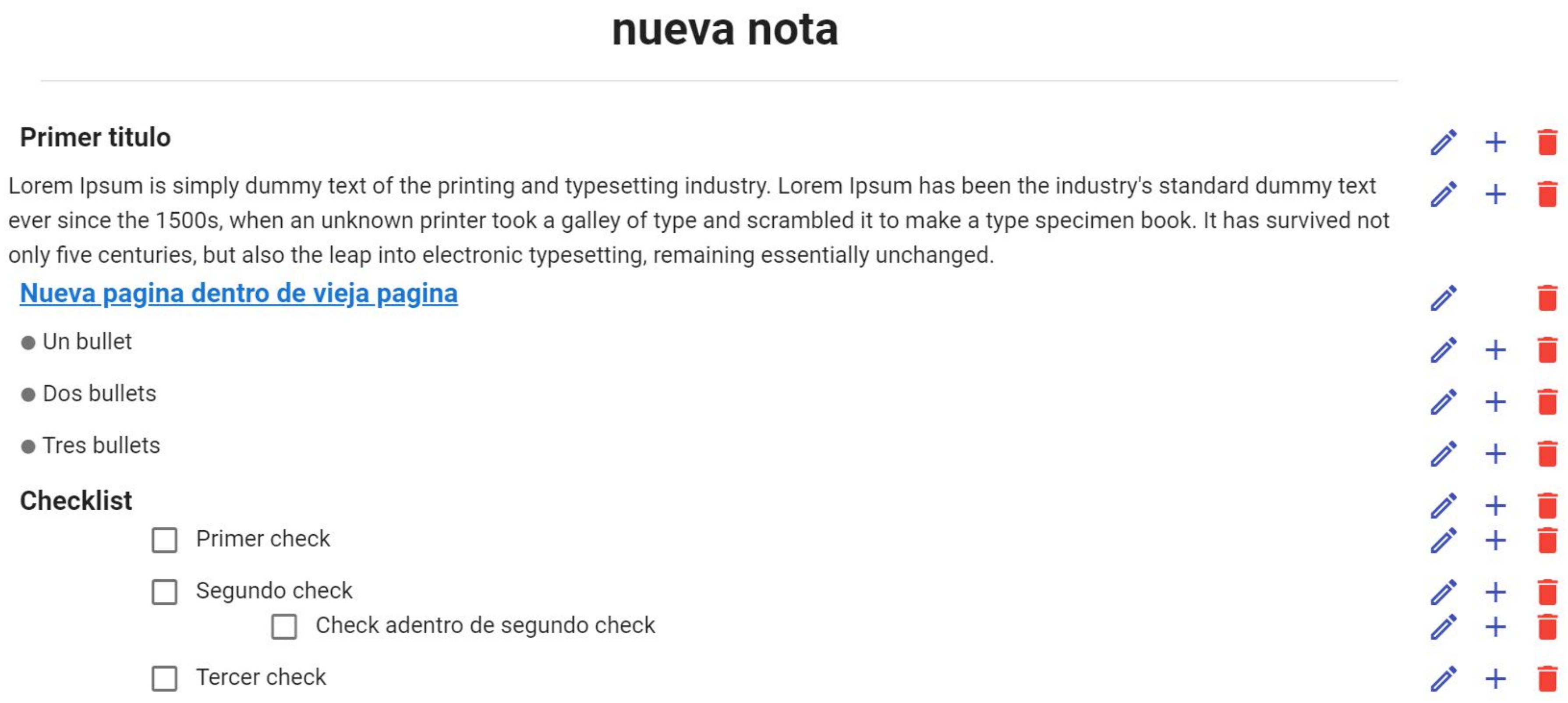
- Un bullet
- Dos bullets
- Tres bullets



Checklist

- [✓] Primer check
- [ ] Segundo check
  - [ ] Check adentro de segundo check
  - [✓] Segundo check dentro del check
- [ ] Tercer check

ESTRUCTURA DE LOS DOCUMENTOS



nueva nota

Primer titulo

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.

[Nueva pagina dentro de vieja pagina]

- Un bullet
- Dos bullets
- Tres bullets

Checklist

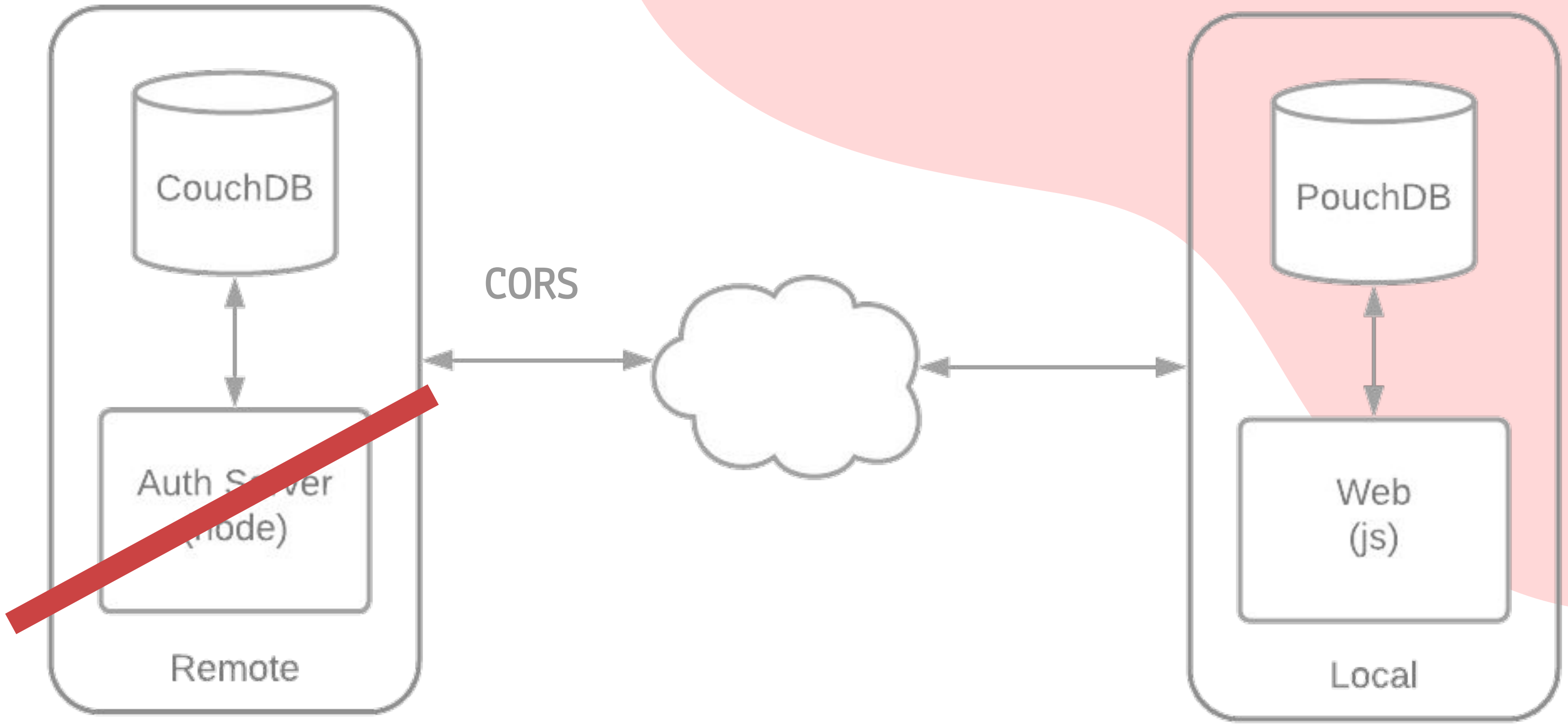
- [ ] Primer check
- [ ] Segundo check
    - [ ] Check adentro de segundo check
- [ ] Tercer check

04

ARQUITECTURA

Cómo comunicar a PouchDB con CouchDB para recuperar mis notas

ARQUITECTURA



BACKEND

CouchDB

- Validar credenciales
- Generar tokens
- Manejar usuarios
- Almacenar las notas y usuarios
- Sincronización de los datos
- Una db por usuario

FRONTEND

Web

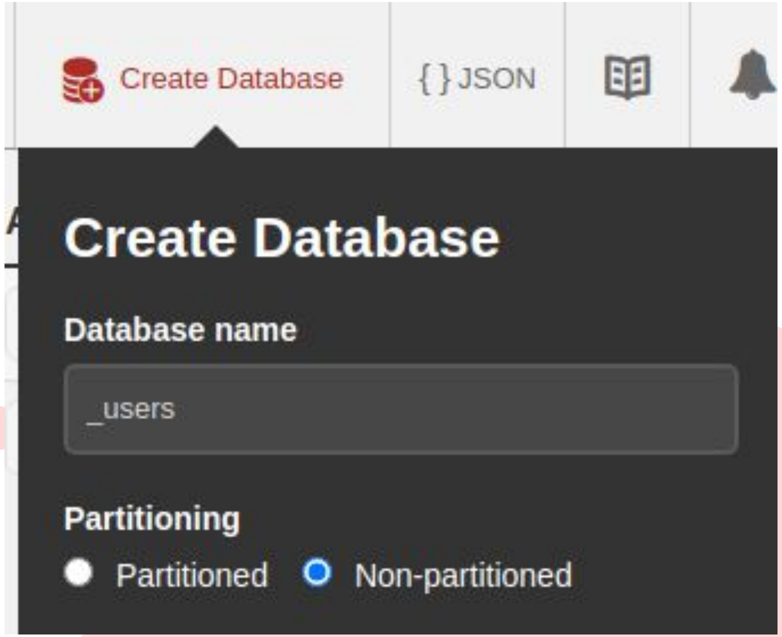
- Manejo de login y credenciales
- Vue.js ⟶ Librería: PouchVue

PouchDB

- Sincronización de la información
- Replica local
- Trabajo offline

AUTENTICACIÓN

1. Crear DB _users



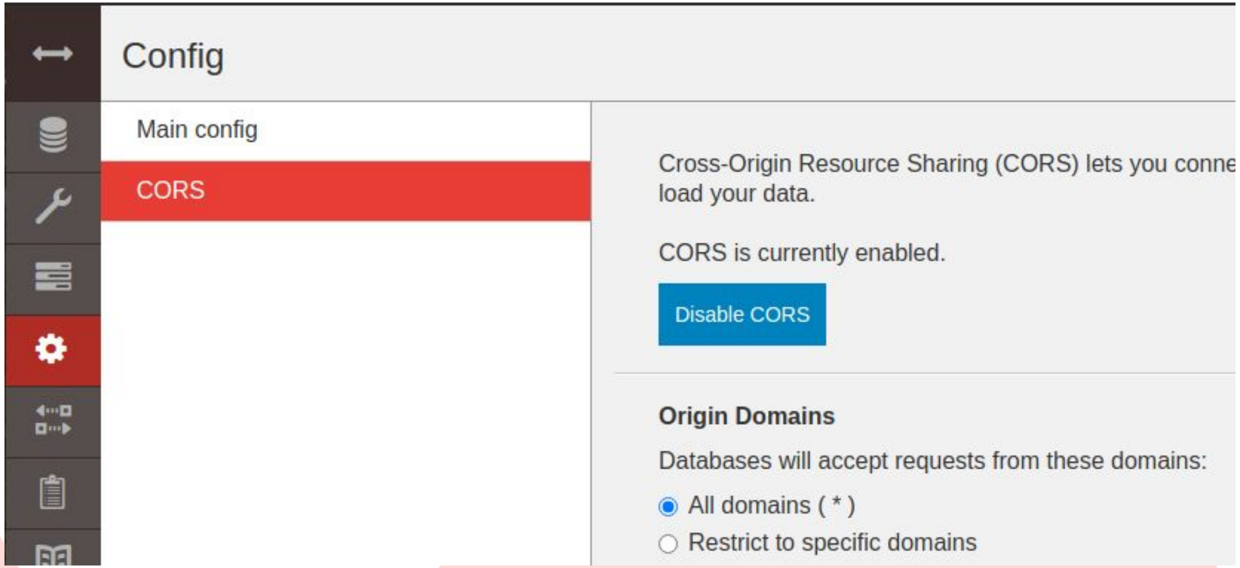
Create Database

Database name: _users

Partitioning: ( ) Partitioned [✓ Non-partitioned]

AUTENTICACIÓN

2. Habilitar CORS



Cross-Origin Resource Sharing (CORS) lets you conne
load your data.

CORS is currently enabled.

Origin Domains

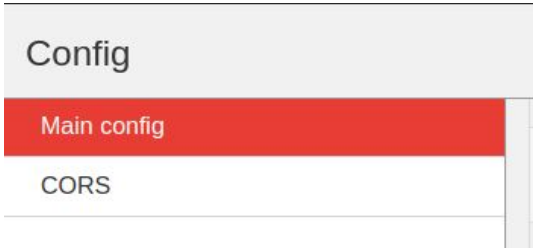
Databases will accept requests from these domains:

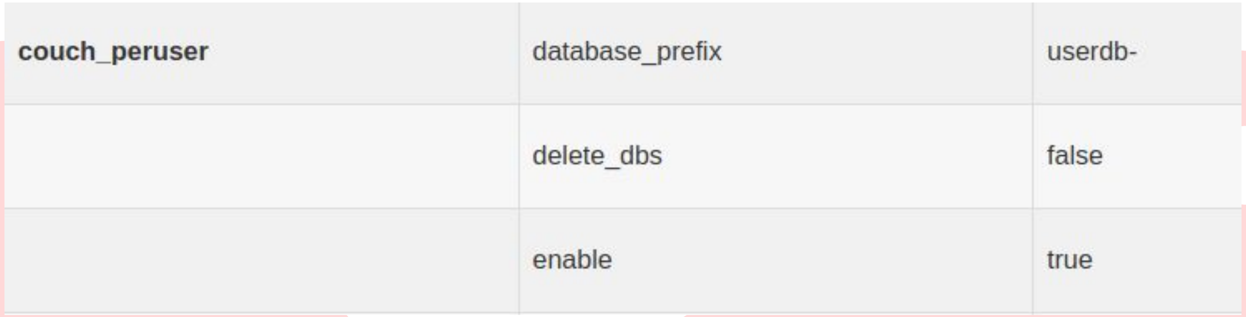
[✓ All domains (*)]
Restrict to specific domains

Permite conectarse a servers remotos directamente desde el browser

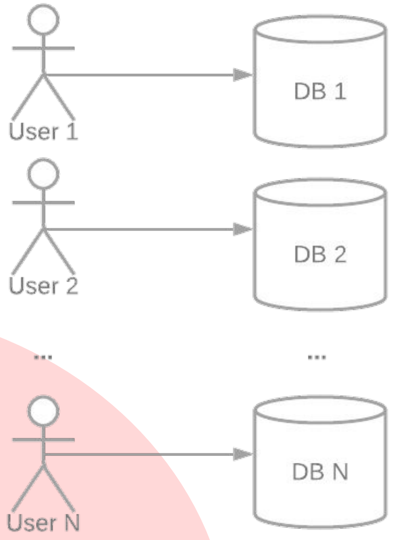
AUTENTICACIÓN

3. Habilitar couch_peruser



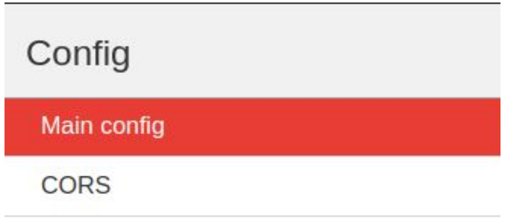


couch_peruser	database_prefix	userdb-
	delete_dbs	false
	enable	true



AUTENTICACIÓN

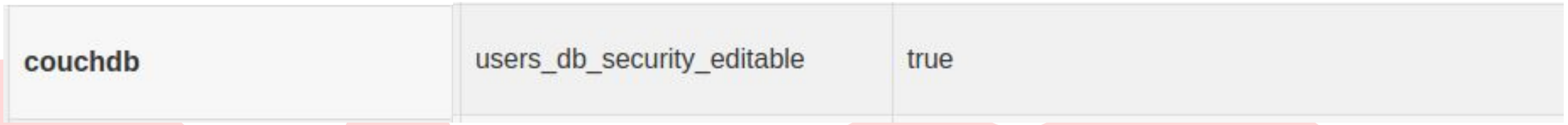
4. Habilitar seguridad editable


Config

Main config

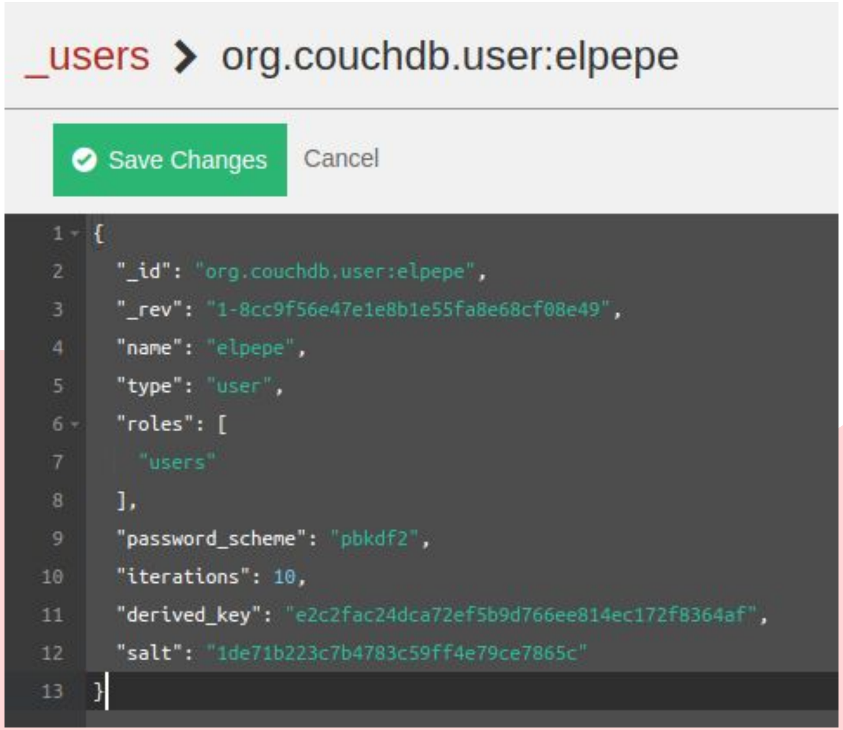
CORS

Permite asignar
roles a los usuarios

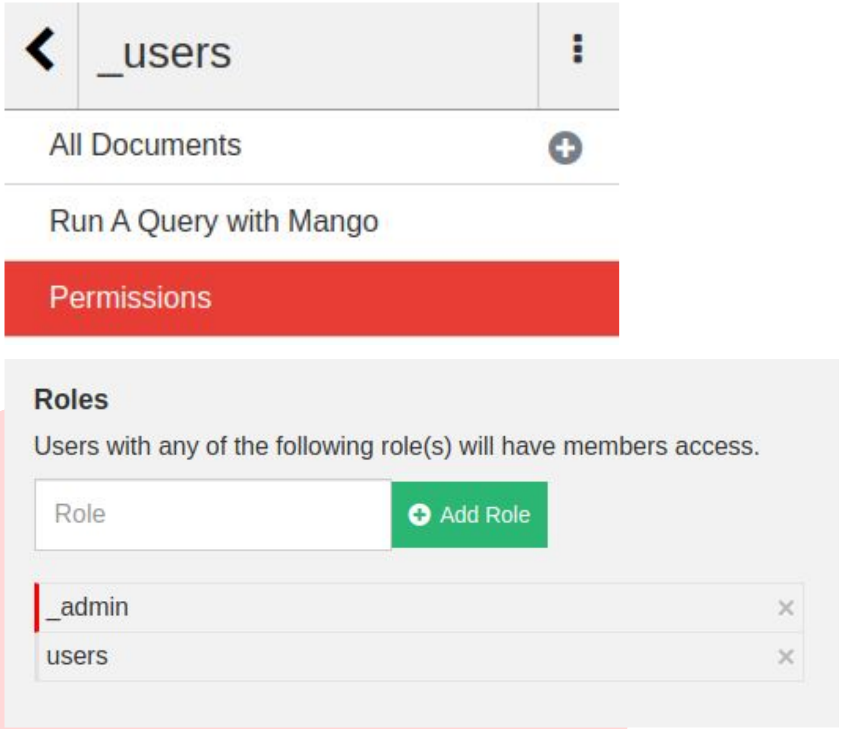

couchdb	users_db_security_editable	true

AUTENTICACIÓN

5. Crear usuario y asignar rol


_users > org.couchdb.user:elpepe

```
{
  "_id": "org.couchdb.user:elpepe",
  "_rev": "1-8cc9f56e47e1e8b1e55fa8e68cf08e49",
  "name": "elpepe",
  "type": "user",
  "roles": [
    "users"
  ],
  "password_scheme": "pbkdf2",
  "iterations": 10,
  "derived_key": "e2c2fac24dca72ef5b9d766ee814ec172f8364af",
  "salt": "1de71b223c7b4783c59ff4e79ce7865c"
}
```



Roles

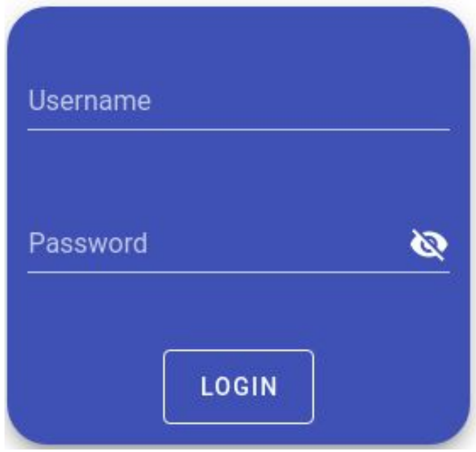
Users with any of the following role(s) will have members access.

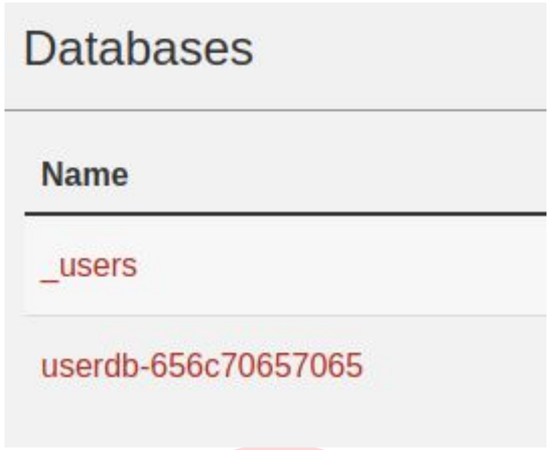
_admin

users

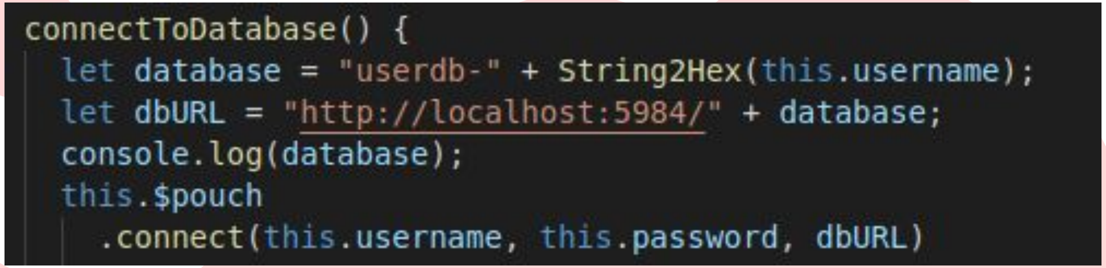
AUTENTICACIÓN

6. Login




Databases

Name
_users
userdb-656c70657065



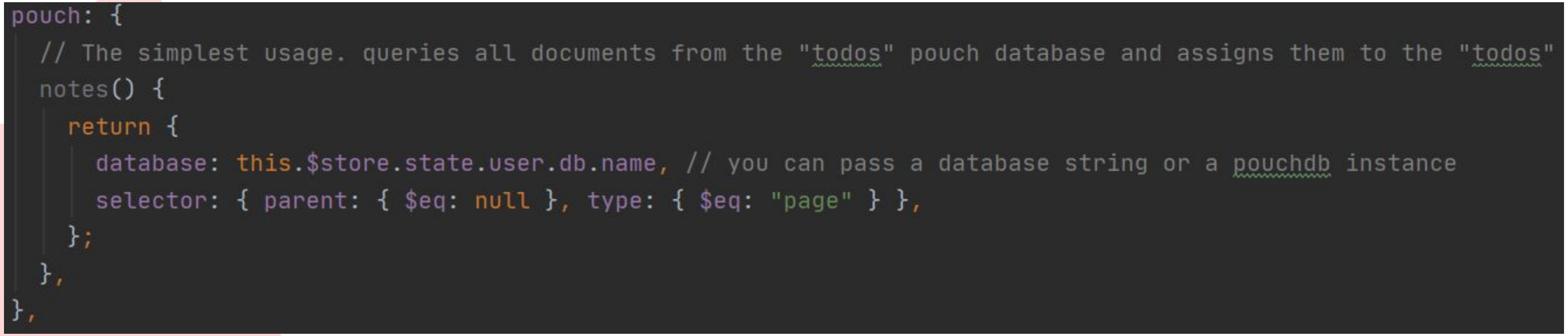
```
connectToDatabase() {
    let database = "userdb-" + String2Hex(this.username);
    let dbURL = "http://localhost:5984/" + database;
    console.log(database);
    this.$pouch
      .connect(this.username, this.password, dbURL)
```

05

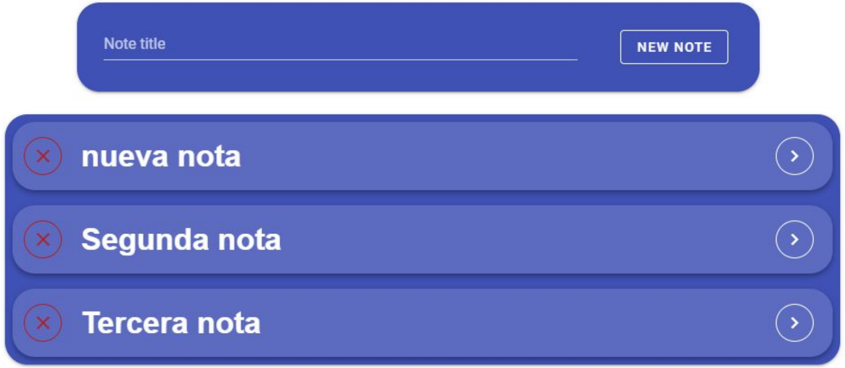
QUERIES

Posibles operaciones CRUD a utilizar

OBTENER PÁGINAS SIN PÁGINA PADRE



```
pouch: {
  // The simplest usage. queries all documents from the "todos" pouch database and assigns them to the "todos"
  notes() {
    return {
      database: this.$store.state.user.db.name, // you can pass a database string or a pouchdb instance
      selector: { parent: { $eq: null }, type: { $eq: "page" } },
    };
  },
},
```



Note title

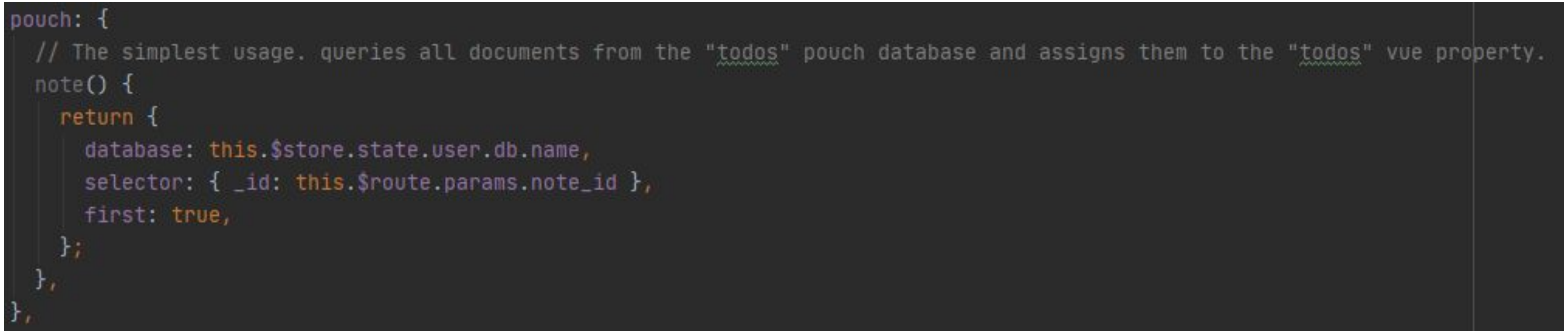
NEW NOTE

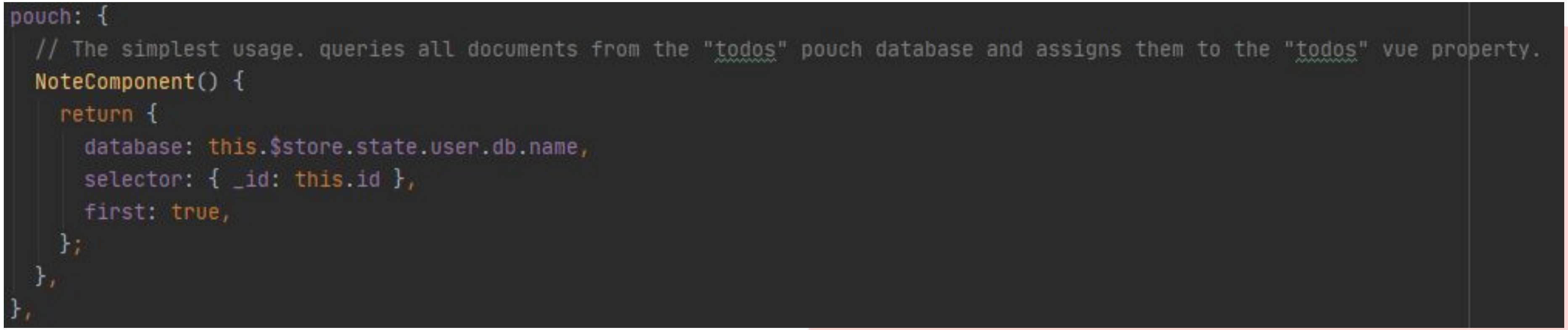
nueva nota

Segunda nota

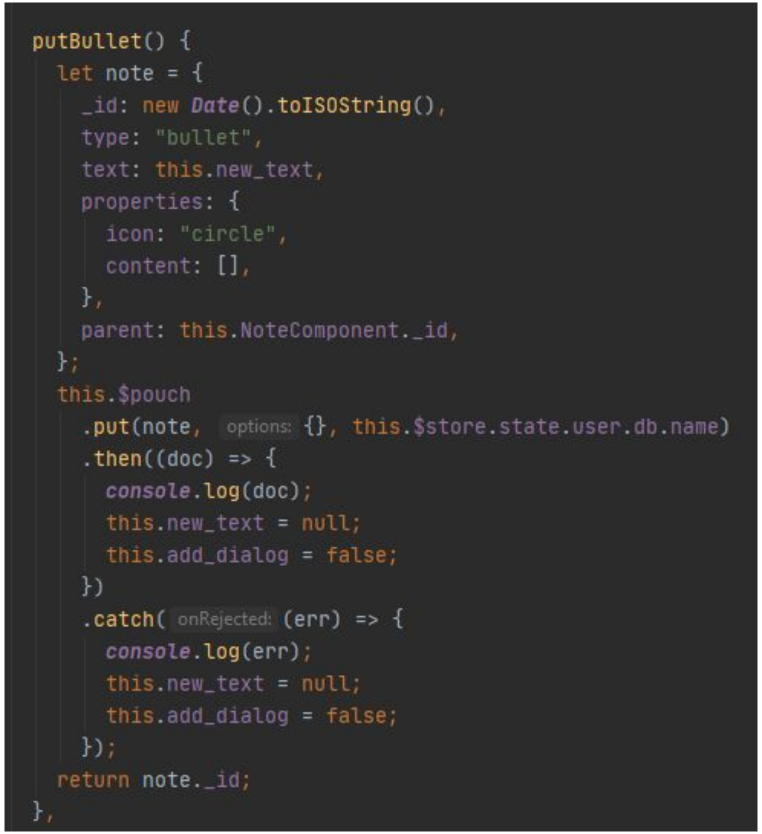
Tercera nota

OBTENER UNA NOTA POR SU ID

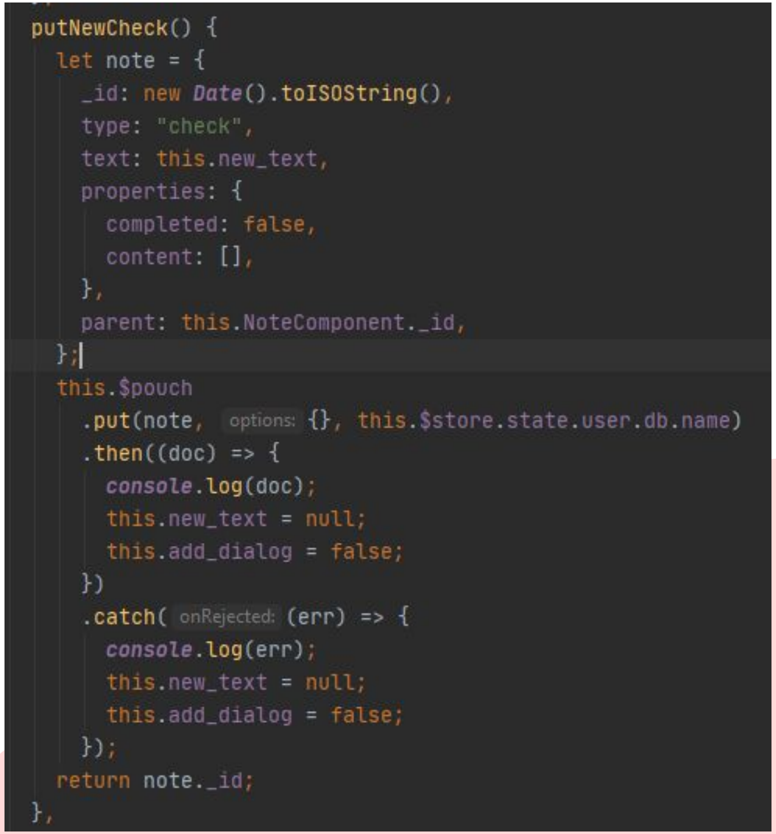

```
pouch: {
  // The simplest usage. queries all documents from the "todos" pouch database and assigns them to the "todos" vue property.
  note() {
    return {
      database: this.$store.state.user.db.name,
      selector: { _id: this.$route.params.note_id },
      first: true,
    };
  },
},
```


```
pouch: {
  // The simplest usage. queries all documents from the "todos" pouch database and assigns them to the "todos" vue property.
  NoteComponent() {
    return {
      database: this.$store.state.user.db.name,
      selector: { _id: this.id },
      first: true,
    };
  },
},
```

CREAR UN NUEVO BULLET VS NUEVO CHECK

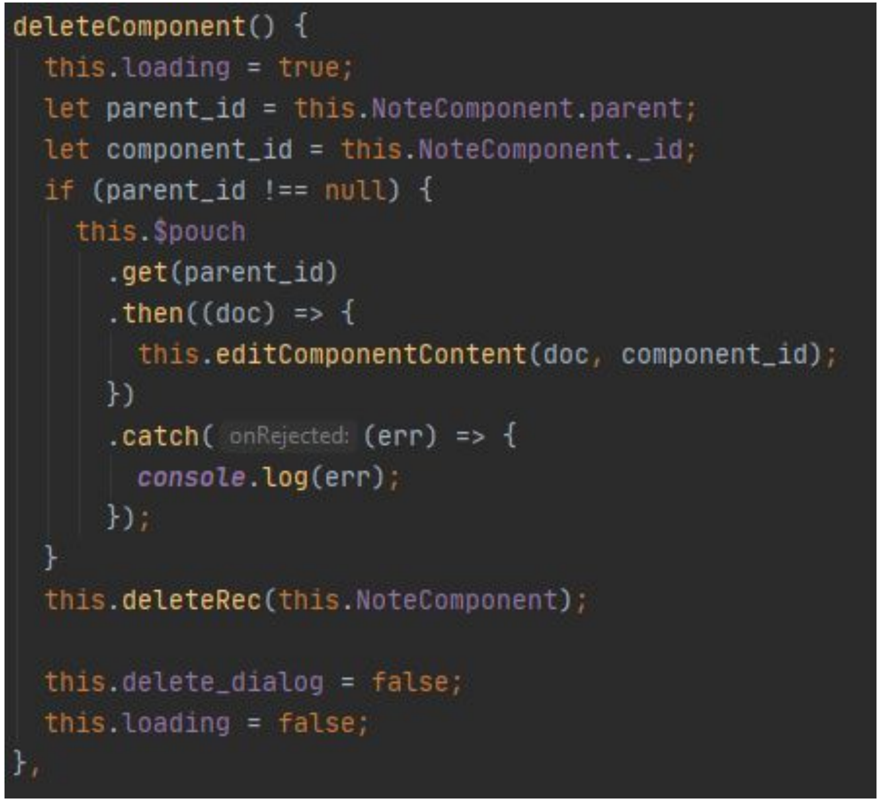


```
putBullet() {
  let note = {
    _id: new Date().toISOString(),
    type: "bullet",
    text: this.new_text,
    properties: {
      icon: "circle",
      content: [],
    },
    parent: this.NoteComponent._id,
  };
  this.$pouch
    .put(note,  options: {}, this.$store.state.user.db.name)
    .then((doc) => {
      console.log(doc);
      this.new_text = null;
      this.add_dialog = false;
    })
    .catch( onRejected: (err) => {
      console.log(err);
      this.new_text = null;
      this.add_dialog = false;
    });
  return note._id;
},
```



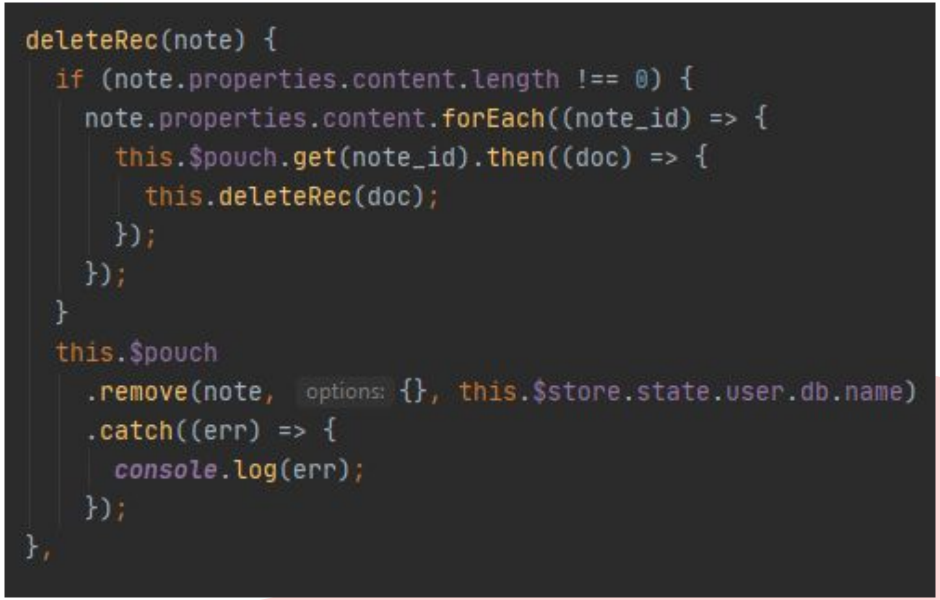
```
putNewCheck() {
  let note = {
    _id: new Date().toISOString(),
    type: "check",
    text: this.new_text,
    properties: {
      completed: false,
      content: [],
    },
    parent: this.NoteComponent._id,
  };
  this.$pouch
    .put(note,  options: {}, this.$store.state.user.db.name)
    .then((doc) => {
      console.log(doc);
      this.new_text = null;
      this.add_dialog = false;
    })
    .catch( onRejected: (err) => {
      console.log(err);
      this.new_text = null;
      this.add_dialog = false;
    });
  return note._id;
},
```

BORRAR UNA NOTA Y SUS HIJOS RECURSIVAMENTE



```
deleteComponent() {
  this.loading = true;
  let parent_id = this.NoteComponent.parent;
  let component_id = this.NoteComponent._id;
  if (parent_id !== null) {
    this.$pouch
      .get(parent_id)
      .then((doc) => {
        this.editComponentContent(doc, component_id);
      })
      .catch( onRejected: (err) => {
        console.log(err);
      });
  }
  this.deleteRec(this.NoteComponent);

  this.delete_dialog = false;
  this.loading = false;
},
```



```
deleteRec(note) {
  if (note.properties.content.length !== 0) {
    note.properties.content.forEach((note_id) => {
      this.$pouch.get(note_id).then((doc) => {
        this.deleteRec(doc);
      });
    });
  }
  this.$pouch
    .remove(note,  options: {}, this.$store.state.user.db.name)
    .catch((err) => {
      console.log(err);
    });
},
```

DEMO TIME!