

# Let's build a point of sale system, using a Vue or two!

---

Ramón Huidobro

[ramonh.dev/point-of-sale-vue-2021.pdf](https://ramonh.dev/point-of-sale-vue-2021.pdf)



[@hola\\_soy\\_milk](#)

# I'm Ramón.

he/him

From Chile, living in Austria

Developer advocate at CodeSee

Software development contractor

Ruby, JS, Rust

Community member

Mozilla tech speaker

Kids' coding coach

# Takeaway.io

# Takeaway.io

Catering

# Takeaway.io

Catering

Cafeteria lunch for  
workers in the  
industrial areas

# Takeaway.io

Catering

Cafeteria lunch for  
workers in the  
industrial areas

Restaurant

# Takeaway.io

Catering

Cafeteria lunch for  
workers in the  
industrial areas

Restaurant













“Please develop a new point of sale system for us”

---

...But why?

---

At the end of the month...



So let's write our own, then!

---



# Project Kassa-dilla

1. Scan chip card
2. Confirm orders
3. Print receipt
4. Nom

“Oh, and could you be done  
in 3 weeks?”

---

The client

[@hola\\_soy\\_milk](#)



# Let's make a web app

- Hardware
-

@hola\_soy\_milk



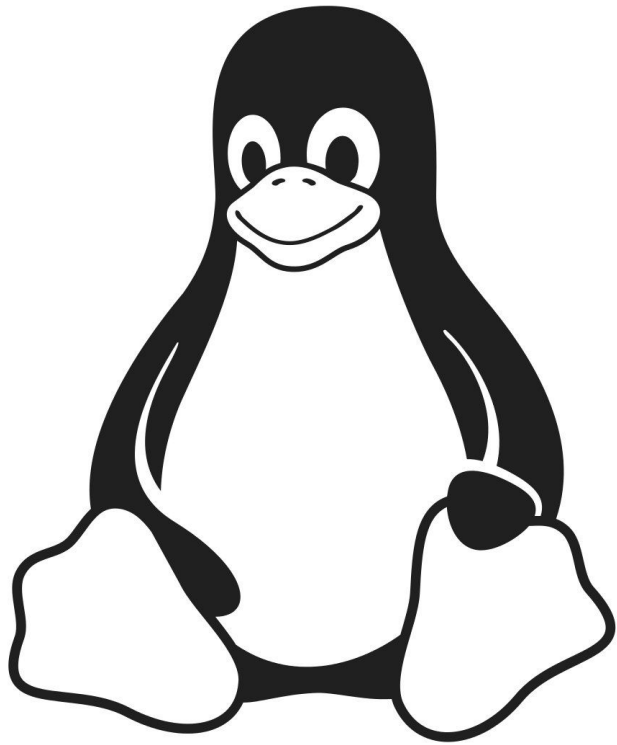
@hola\_soy\_milk



@hola\_soy\_milk

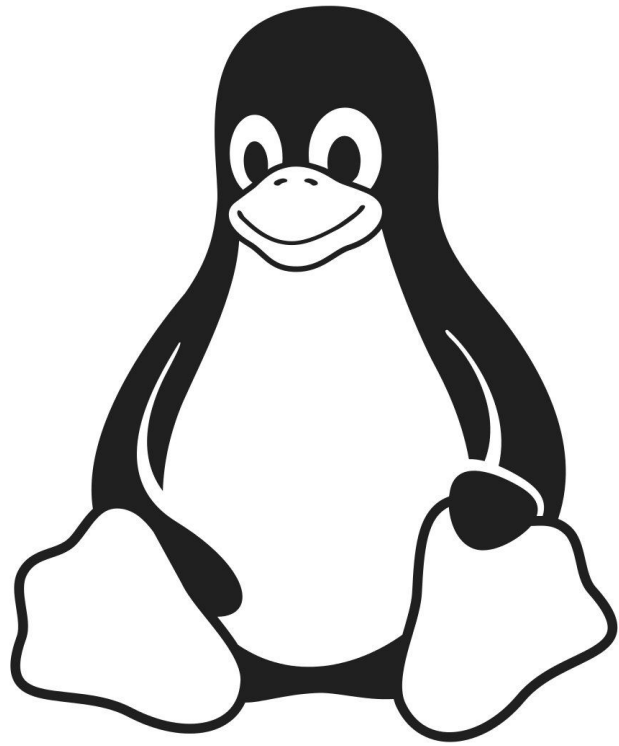


[@hola\\_soy\\_milk](#)

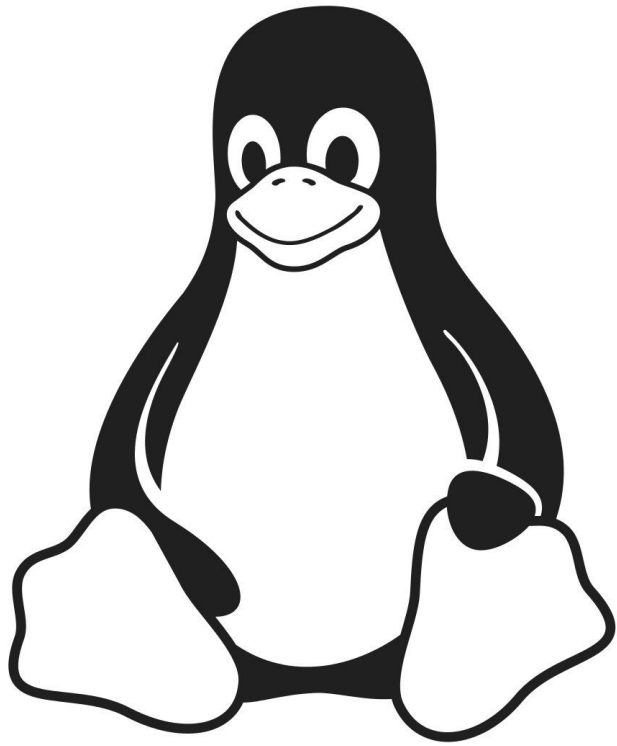




@hola\_soy\_milk



@hola\_soy\_milk



ЕЖКИ

GRUBER

@hola\_soy\_milk



# Let's make a web app

- Hardware
- Our vue.js app



# What a vue!

bundle exec rails webpacker:install:vue

```
{
  "dependencies": {
    "vue": "^2.6.10",
    "vue-loader": "14.2.2",
    "vue-template-compiler": "^2.6.10",
    "vue-turbolinks": "^2.0.4"
  },
  "devDependencies": {
    "webpack-cli": "^3.3.6",
    "webpack-dev-server": "2.11.5"
  }
}
```

```
<Root>  
  <PointOfSale>  
    <UserLookup>  
    <Menu>  
    <Checkout>  
      <ShoppingCart>  
      <Total>  
      <Submit>
```

# Let's make a web app

- ~~Hardware~~
  - ~~Our vue.js app~~
  - <UserLookup>
-







4962 ↩

# bootstrap-select

The jQuery plugin that brings select elements into the 21st century with intuitive multiselection, searching, and much more. Now with Bootstrap 4 support.

 Download (v1.13.9)



Star

8,772



Fork

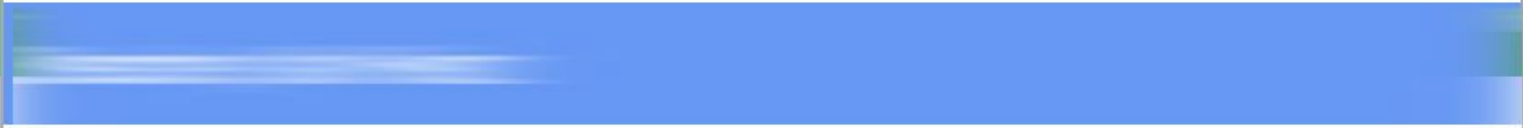
2,570

## Getting Started

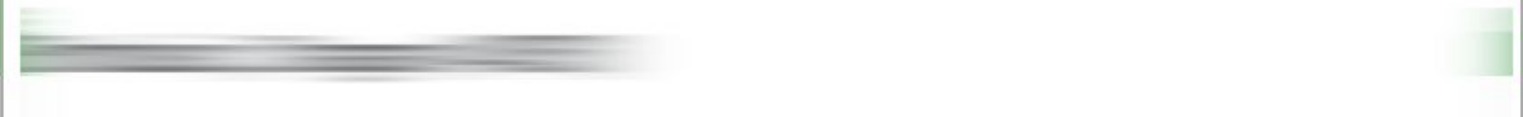
### Quick start

Bootstrap-select requires jQuery v1.9.1+, Bootstrap's dropdown.js component, and Bootstrap's CSS. If you're not already using Bootstrap in your project, a precompiled version of the Bootstrap v3.3.7 minimum requirements can be downloaded [here](#). If using bootstrap-select with Bootstrap v4+, you'll also need Popper.js. For all of Bootstrap v4's requirements, see [Getting started](#). A precompiled version of the requirements will be made available in an upcoming release of bootstrap-select.

Briggs, Abida (4962)

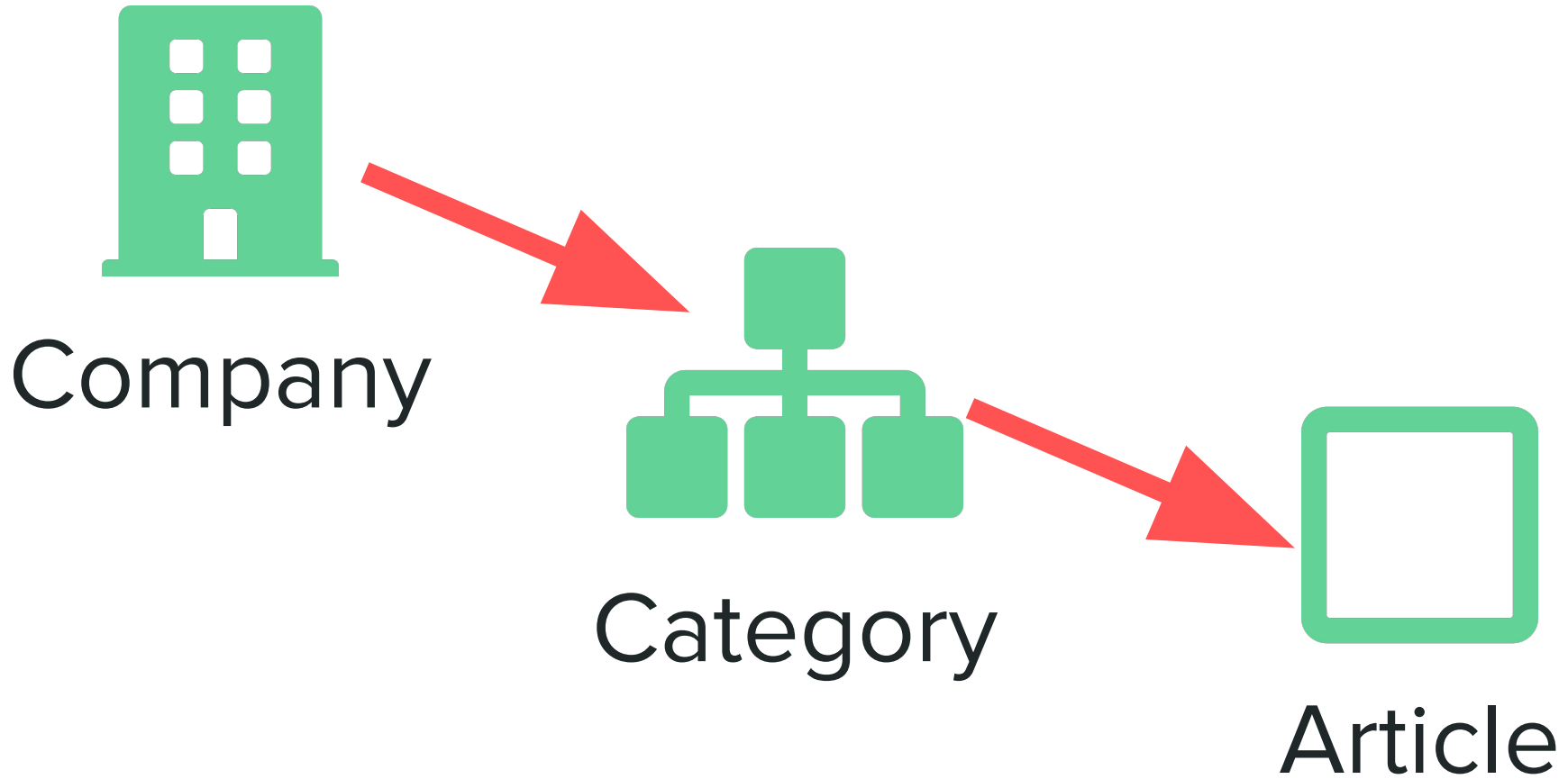


Briggs, Abida (4962)



# Let's make a web app

- ~~Hardware~~
  - ~~Our vue.js app~~
  - ~~<UserLookup>~~
  - <Menu>
-



Menü 1 - Fleisch

Menü Budget

Salat 2 - Vegetarisch

Suppe zum Menü

Salatbuffet zum Menü

Dessert

Menü 2 - Vegetarisch

Salat 1 - Fleisch

Schnitzel

Suppe solo

Salatbuffet solo

Gerichte

Eis



Hold on, where's all this data coming from?

---

[Installation](#)[What is Vuex?](#)[Getting Started](#)

## Core Concepts

[State](#)[Getters](#)[Mutations](#)[Actions](#)[Modules](#)[Application Structure](#)[Plugins](#)[Strict Mode](#)[Form Handling](#)[Testing](#)[Hot Reloading](#)

# What is Vuex?



Vuex is a **state management pattern + library** for Vue.js applications. It serves as a centralized store for all the components in an application, with rules ensuring that the state can only be mutated in a predictable fashion. It also integrates with Vue's official [devtools extension](#) to provide advanced features such as zero-config time-travel debugging and state snapshot export / import.

## What is a "State Management Pattern"?

@hola\_soy\_milk



```
import Vue from 'vue/dist/vue.esm';
import Vuex from 'vuex';
import createMutationsSharer from "vuex-shared-mutations";
Vue.use(Vuex);

import ArticleStore from './stores/article_store';
import OrderItemStore from './stores/order_item_store';
import UserStore from './stores/user_store';
import OrderStore from './stores/order_store';
import PreorderStore from './stores/preorder_store';

const store = new Vuex.Store({
  modules: {
    ArticleStore,
    UserStore,
    OrderStore,
    PreorderStore,
    OrderItemStore,
  }
});

export default store;
```

## vuex/index.js

```
const ArticleStore = {
  namespaced: true,
  state: {
    articles: [],
    categories: [],
    category: {}
  },
  mutations: {
    one(state, data) {
      state.category = data
      return state;
    },
    many(state, data) {
      state.categories = data;
      let articleSets = state.categories.map((cat) => { return cat.pos_articles })
      state.articles = Array.prototype.concat.apply([], articleSets)
      state.category = state.categories[0]
      return state;
    }
  },
  actions: {
    index(context, query) {
      $.ajax({
        url: `articles`,
        type: 'get',
        data: query,
        success: function(data) {
          context.commit('many', data)
        }
      })
    }
  }
};

export default ArticleStore;
```

article\_store.js

```
<template>
  <div>
    <div class="articles container pt-3">
      <div class="row">
        <div class="col-6 mb-3 text-center" v-on:click="addArticle" v-for="article in category.pos_articles" :key="article.id">
          <button type="button" :data-article-id="article.id" class="btn btn-success w-75 ml-auto mr-auto">
            {{ article.name }}
          </button>
        </div>
      </div>
    </div>
    <div class="btn-group" role="group">
      <button type="button" v-on:click="selectCategory" v-for="category in categories" :key="category.id" :data-category-id="
category.id" :id="idForCategoryButton(category)" :class="buttonClass(category)">
        {{ category.name }}
      </button>
    </div>
  </div>
</template>
```

menu.vue

```
<template>
  <div>
    <div class="articles container pt-3">
      <div class="row">
        <div class="col-6 mb-3 text-center" v-on:click="addArticle" v-for="article in category.pos_articles" :key="article.id">
          <button type="button" :data-article-id="article.id" class="btn btn-success w-75 ml-auto mr-auto">
            {{ article.name }}
          </button>
        </div>
      </div>
    </div>
    <div class="btn-group" role="group">
      <button type="button" v-on:click="selectCategory" v-for="category in categories" :key="category.id" :data-category-id="
category.id" :id="idForCategoryButton(category)" :class="buttonClass(category)">
        {{ category.name }}
      </button>
    </div>
  </div>
</template>
```

```
addArticle: function (event) {
  if (event) {
    let article = this.articles.find((article) => {
      return article.id === event.target.getAttribute('data-article-id')
    })
    this.$store.dispatch('OrderItemStore/add', {
      pos_article_id: article.id,
      name: article.name,
      subsidy_cents: article.subsidisable_amount_cents,
      net_cents: article.net_cents,
      gross_cents: article.gross_cents,
      vat_cents: article.vat_cents,
    })
  }
}
```



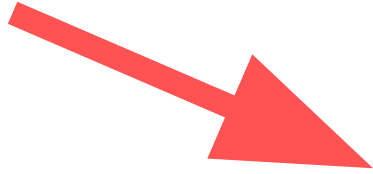
# Let's make a web app

- Hardware
  - Our vue.js app
  - <UserLookup>
  - <Menu>
  - <ShoppingCart>
-

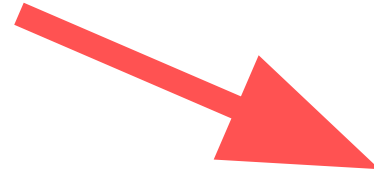
```
<Root>  
  <PointOfSale>  
    <UserLookup>  
    <Menu>  
    <Checkout>  
      <ShoppingCart>  
      <Total>  
      <Submit>
```



User



Order



Order  
Items

Cornetto	2,50 €	Entfernen
Magnum	2,60 €	Entfernen
Calippo	1,80 €	Entfernen

```
<template>
  <div id="shopping-cart" class="pt-1 pb-1 shopping-cart bg-light">
    <table class="table">
      <tr v-for="item in orderItems" class="animated fadeInDown">
        <td >{{ item.name }}</td>
        <td >{{ formatAmount(item.gross_cents) }}</td>
        <td>
          <button v-on:click="removeOrderItem" type="button" :data-item-name="item.name" class="
remove btn btn-outline-danger ml-auto mr-auto">
            Entfernen
          </button>
        </td>
      </tr>
    </table>
  </div>
</template>
```

shopping\_cart.vue

```
removeOrderItem: function (event) {
  if (event) {
    let orderItem = this.orderItems.find((orderItem) => {
      return orderItem.name === event.target.getAttribute('data-item-name')
    })
    console.log(JSON.stringify(orderItem))
    this.$store.dispatch('OrderItemStore/set', this.$store.state.OrderItemStore.orderItems.filter(function(e) { return e !==
orderItem })))
  }
}
```

# Let's make a web app

- Hardware
  - Our vue.js app
  - ~~<UserLookup>~~
  - ~~<Menu>~~
  - ~~<ShoppingCart>~~
  - <Total>
-

```
let amount = this.orderItems.reduce( (previous, item) => {  
  return previous + item.gross_cents  
}, 0)
```



# Let's make a web app

- Hardware
  - Our vue.js app
  - `<UserLookup>`
  - `<Menu>`
  - `<ShoppingCart>`
  - `<Total>`
  - Sending the order off
-

```
submitOrder: async function(event) {
  const overlay = document.querySelector('#loading')
  overlay.classList.remove('hidden')
  let order = {
    order: {
      user_id: this.$store.state.UserStore.userId,
      pos_order_items_attributes: this.$store.state.OrderItemStore.orderItems
    }
  }
  await this.$store.dispatch('OrderStore/create', order)
  this.$store.dispatch('OrderItemStore/reset')
  element.classList.add('animated', 'rubberBand')
  overlay.classList.add('hidden')
},
```

# Let's make a web app

- ~~Hardware~~
  - ~~Our vue.js app~~
  - ~~<UserLookup>~~
  - ~~<Menu>~~
  - ~~<ShoppingCart>~~
  - ~~<Total>~~
  - ~~Sending the order off~~
  - Receipts
-



## Command Descriptions

### Command Notation

<b>[Name]</b>	The name of the control command.
<b>[Format]</b>	The code sequence. In this description, < > H denotes hexadecimal numbers, < > denotes decimal numbers and < > B denotes binary numbers. [ ] k indicates the contents of the [ ] should be repeated k times.
<b>[Range]</b>	The allowable range for the arguments.
<b>[Description]</b>	Description of the command function.
<b>[Details]</b>	If necessary provides important information on setting and using the printer command.
<b>[Default]</b>	The default values for the commands.
<b>[Reference]</b>	List related commands.
<b>[Example]</b>	Example of using the commands.

The numbers denoted by <>H is hexadecimal.







The numbers denoted by <>B is binary.

## Print Commands

The WTP series supports the following commands for printing characters and advancing paper.

### HT

<b>[Name]</b>	Horizontal tab
<b>[Format]</b>	ASCII HT

 <a href="#">.ruby-version</a>	add attr_reader for data to Escpos::Printer, update ruby version	6 months ago
 <a href="#">Gemfile</a>	Initial escpos implementation	4 years ago
 <a href="#">README.md</a>	Update README.md	6 months ago
 <a href="#">Rakefile</a>	cleanup, add CI	2 years ago
 <a href="#">config</a>	Initial escpos implementation	4 years ago
 <a href="#">escpos.gemspec</a>	cleanup, add CI	2 years ago

@hola\_soy\_milk

## README.md

### Build Status

# Escpos

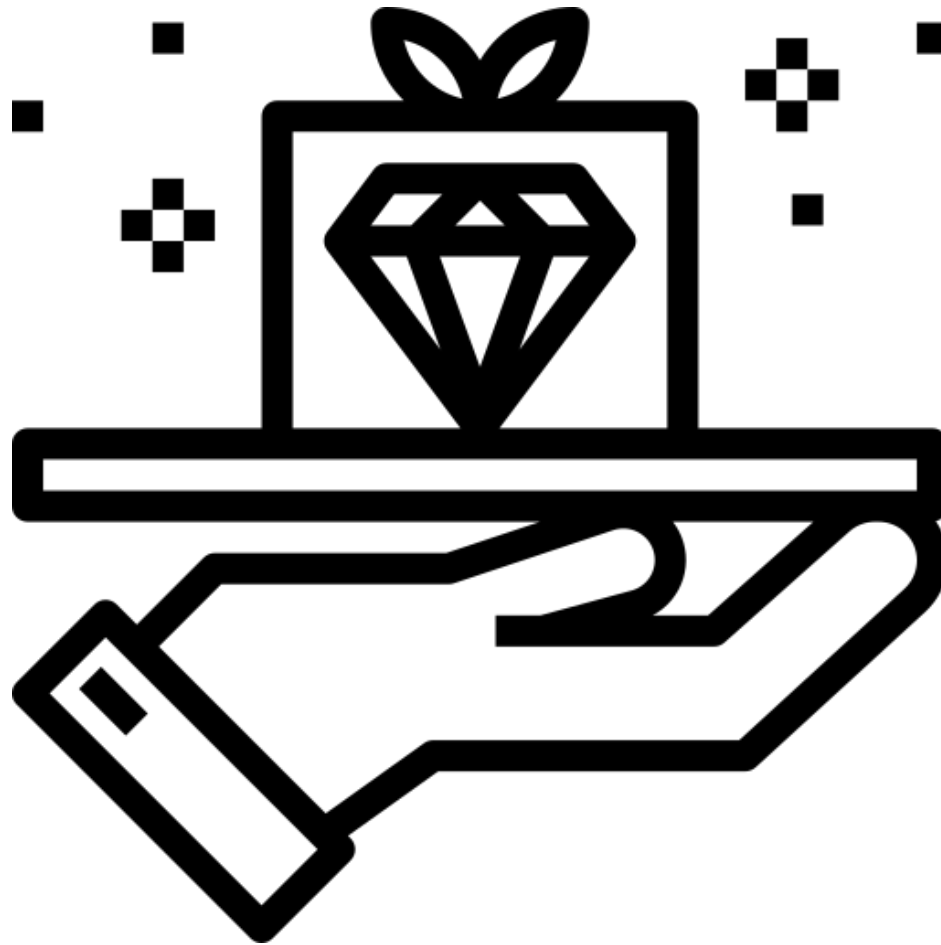
A ruby implementation of ESC/POS (thermal) printer command specification.

## Installation

Add this line to your application's Gemfile:

```
gem 'escpos'
```

Method name	Description
<code>text</code>	Normal text formatting
<code>encoding, set_encoding, set_printer_encoding</code>	Set printer encoding (see example below)
<code>encode</code>	Encode text for the printer (see example below)
<code>double_height</code>	Double height text
<code>quad_text, big, title, header, double_width_double_height, double_height_double_width</code>	Double width & Double height text
<code>double_width</code>	Double width text
<code>underline, u</code>	Underlined text
<code>underline2, u2</code>	Stronger underlined text
<code>bold, b</code>	Bold text
<code>left</code>	Align to left
<code>right</code>	Align to right
<code>center</code>	Align to center
<code>invert, inverted</code>	Color inverted text
<code>black, default_color, color_black, black_color</code>	Default Color (Usually black)
<code>red, alt_color, alternative_color, color_red, red_color</code>	Alternative Color (Usually Red)
<code>barcode</code>	Print barcode (see example below)
<code>partial_cut</code>	Partially cut the paper (may not be available on all devices)
<code>cut</code>	Fully cut the paper (may not be available on all devices)





```
orders_controller.rb  
if @order.save!  
  doc = EscDocument.new  
  PointOfSale::OrderEscBuilder.new(@order, doc).print  
  render json: {receipt: Base64.encode64(doc.to_escpos)}  
end
```

```
def print_vat
  doc.body_text_underlined(doc.start_aligning_left('Im Betrag enthaltene MwSt:'))
  format = "%-7.5s%2s%10s%10s%10s"
  row = sprintf(format, '0,0%', '=', '0,00', 'Netto:', '0,00')
  doc.body_text(row)
  vat = format_currency(order.total_vat.to_f, false)
  net = format_currency(order.total_net.to_f, false)
  row = sprintf(format, '10,0%', '=', "#{vat}", 'Netto:', net)
  doc.body_text(row)
  row = sprintf(format, '20,0%', '=', '0,00', 'Netto:', '0,00')
  doc.body_text(row)
end
```

order\_esc\_builder.rb

```
const OrderStore = {
  namespace: true,
  state: {
    receipt: {},
  },
  mutations: {
    created(state, data) {
      state.order = {}
      state.orders = []
      state.receipt = data.receipt
      return state;
    },
  },
  actions: {
    async create(context, query) {
      await $.ajax({
        url: `orders`,
        type: 'post',
        data: JSON.stringify(query),
        dataType: 'json',
        contentType: 'application/json',
        success: async function(data) {
          await context.commit('created', data)
        }
      })
    },
  },
};
```

order\_store.js

```
export default OrderStore;
```

```
submitOrder: async function(event) {
  const overlay = document.querySelector('#loading')
  overlay.classList.remove('hidden')
  let order = {
    order: {
      user_id: this.$store.state.UserStore.userId,
      pos_order_items_attributes: this.$store.state.OrderItemStore.orderItems
    }
  }
  await this.$store.dispatch('OrderStore/create', order)
  let receipt = this.$store.state.OrderStore.receipt;
  axios.post("localhost:45612", receipt)
  this.$store.dispatch('OrderItemStore/reset')
  element.classList.add('animated', 'rubberBand')
  overlay.classList.add('hidden')
},
},
```

checkout.vue

```
post '/' do
  data = Base64.decode64(request.body.read)

  unless ENV['DEBUG']
    fd = IO.sysopen(device, 'w+')
    printer = IO.new(fd)

    # Split into chunks, otherwise the printer stop printing, somewhere in the
    # vicinity of 8KB
    data.bytes.each_slice(2048) do |slice|
      chunk = slice.pack 'C*'
      printer.puts(chunk)
      printer.flush
    end

    printer.close
  end

  # Log
  puts data
end
```

esc\_printer\_server.rb

BonNr: 2987 Datum: 2019/12/04 10:22:45

## Lieferschein

Bezeichnung	Stk.	Preis	Gesamt
Salatbuffet solo	7	1.000,00	7.000,00
Salatbuffet zum Menü	4	0,70	2,80
Dessert	3	1.000,00	3.000,00
Suppe solo	3	1,10	3,30
Suppe zum Menü	4	0,60	2,40
Salat 2 - Vegetaris.	7	5,20	36,40
Schnitzel	4	5,20	20,80
Salat 1 - Fleisch	4	5,20	20,80
Menü 2 - Vegetarisch	6	5,20	31,20
Menü Budget	3	4,70	14,10
Menü 1 - Fleisch	5	5,20	26,00
<b>Gesamt:</b>			<b>10.157,80</b>

Im Betrag enthaltene MwSt:

0,0%	=	0,00	Netto:	0,00
10,0%	=	1.014,25	Netto:	9.143,55
20,0%	=	0,00	Netto:	0,00

# Let's make a web app

- ~~Hardware~~
  - ~~Our vue.js app~~
  - ~~<UserLookup>~~
  - ~~<Menu>~~
  - ~~<ShoppingCart>~~
  - ~~<Total>~~
  - ~~Sending the order off~~
  - ~~Receipts~~
-

Fortunately, it was done in about 2 weeks!

---





# Problem number 1

---

“Oh wait, but let’s not use that...”

@hola\_soy\_milk





**@hola\_soy\_milk**



Filter by title

Universal Serial Bus (USB)

> New for USB in different versions of Windows

> Concepts for all USB developers

> Building USB devices for Windows

USB Dual Role Driver Stack Architecture

USB host-side drivers in Windows

USB device-side drivers in Windows

▼ Developing Windows applications for USB devices

Overview of developing Windows applications for USB devices

> Writing a UWP app for a USB device

▼ Writing a Windows desktop app for a USB device

Windows desktop app for a USB device

Write a Windows desktop app based on the WinUSB template

How to Access a USB Device by Using

The application uses the **Pipeld** value to identify which pipe to use for data transfer in calls to WinUSB functions, such as [WinUsb\\_ReadPipe](#) (dr@hola\_soy\_milk Requests" section of this topic), so the example stores all three **Pipeld** values for later use.

The following example code gets the speed of the device that is specified by the WinUSB interface handle.

ManagedCPlusPlus

Copy

```
BOOL GetUSBDeviceSpeed(WINUSB_INTERFACE_HANDLE hDeviceHandle, UCHAR* pDeviceSpeed)
{
    if (!pDeviceSpeed || hDeviceHandle==INVALID_HANDLE_VALUE)
    {
        return FALSE;
    }

    BOOL bResult = TRUE;

    ULONG length = sizeof(UCHAR);

    bResult = WinUsb_QueryDeviceInformation(hDeviceHandle, DEVICE_SPEED, &length);
    if(!bResult)
    {
        printf("Error getting device speed: %d.\n", GetLastError());
        goto done;
    }

    if(*pDeviceSpeed == LowSpeed)
    {
```

Filter by title

Universal Serial Bus (USB)

> New for USB in different versions of Windows

> Concepts for all USB developers

> Building USB devices for Windows

USB Dual Role Driver Stack Architecture

USB host-side drivers in Windows

USB device-side drivers in Windows

▼ Developing Windows applications for USB devices

Overview of developing Windows applications for USB devices

> Writing a UWP app for a USB device

▼ Writing a Windows desktop app for a USB device

Windows desktop app for a USB device

Write a Windows desktop app based on the WinUSB template

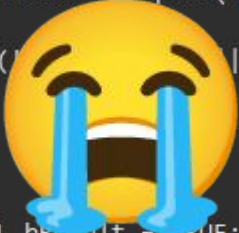
How to Access a USB Device by Using

The application uses the **Pipeld** value to use for data transfer in calls to WinUSB functions, such as [WinUsb\\_ReadPipe](#) (dr@hola\_soy\_milk Requests" section of this topic), so the example stores all three **Pipeld** values for later use.

The following example code gets the speed of the device that is specified by the WinUSB interface handle.

ManagedCplusplus

Copy

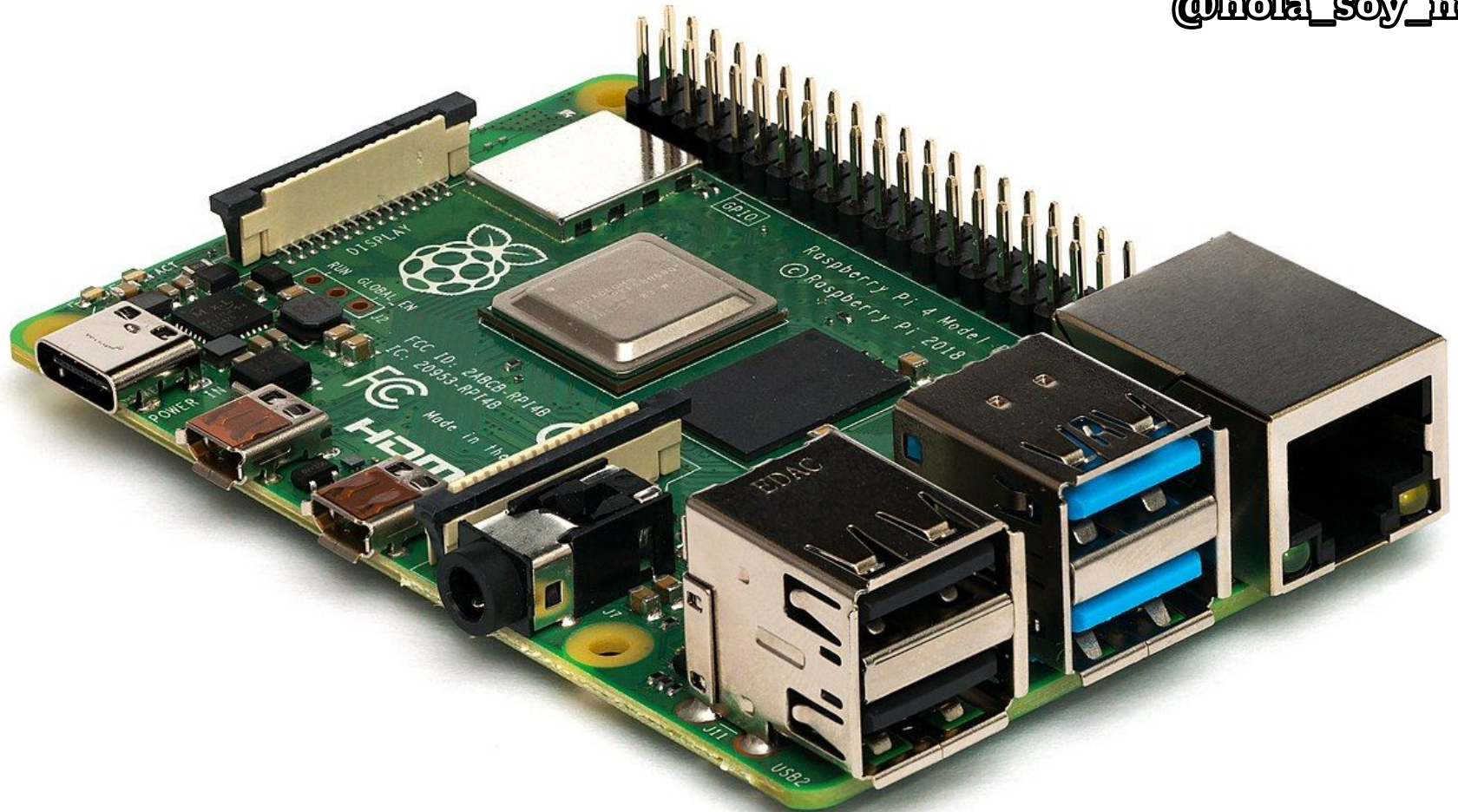
```
BOOL GetUSBDeviceSpeed(WINUSB_INTERFACE_HANDLE hDeviceHandle, UCHAR* pDeviceSpeed)
{
    if (hDeviceHandle == INVALID_HANDLE_VALUE)
    {
        
    }

    BOOL bResult = TRUE;

    ULONG length = sizeof(UCHAR);

    bResult = WinUsb_QueryDeviceInformation(hDeviceHandle, DEVICE_SPEED, &length);
    if(!bResult)
    {
        printf("Error getting device speed: %d.\n", GetLastError());
        goto done;
    }

    if(*pDeviceSpeed == LowSpeed)
    {
```





# Problem number 2

---

“Hey, where’s my food at!”





“Ok cool, I’ll just  
mirror the screens!”

@hola\_soy\_milk

800\*600

1080p



**@hola\_soy\_milk**

800\*600

800\*600



“So how about streaming a portion of one screen to another?”

“So how about streaming a portion of one screen to another?... No.”







“So how about  
streaming the  
DATA?”

2 browser windows,  
you say...



🔍 Search packages

Search

Sign Up

Sign In

Get unlimited public & private packages + package-based permissions with npm Pro. [Get started »](#)

# vuex-shared-mutations

1.0.2 • Public • Published a year ago

[Readme](#)

[Explore](#) BETA

0 Dependencies

4 Dependents

7 Versions

Unable to find a readme for vuex-shared-mutations@1.0.2

## Keywords

vue vuex plugin

### Install

```
> npm i vuex-shared-mutations
```

### Weekly Downloads

15.022



Version

License

```
<Root>
```

```
  <CustomerBasket>
```

```
    <ShoppingCart>
```

```
    <Total>
```

**@hola\_soy\_milk**

0,00 €

```
mutations: {  
  set(state, data) {  
    state.orderItems = data  
    return state;  
  },  
  reset(state) {  
    state.orderItems = []  
    return state;  
  },  
  add(state, data) {  
    state.orderItems.push(data)  
    return state;  
  },  
  one(state, data) {  
    state.orderItem = data  
    return state;  
  },  
  many(state, data) {  
    state.orderItems = data;  
    return state;  
  }  
},
```

# order\_item\_store.js



## Usage

---

```
import createMutationsSharer from "vuex-shared-mutations";

const store = new Vuex.Store({
  // ...
  plugins: [createMutationsSharer({ predicate: ["mutation1", "mutation2"] })]
});
```

<https://github.com/xanf/vuex-shared-mutations>

```
import Vue from 'vue/dist/vue.esm';
import Vuex from 'vuex';
import createMutationsSharer from "vuex-shared-mutations";
Vue.use(Vuex);

import ArticleStore from './stores/article_store';
import OrderItemStore from './stores/order_item_store';
import UserStore from './stores/user_store';
import OrderStore from './stores/order_store';

const store = new Vuex.Store({
  modules: {
    ArticleStore,
    UserStore,
    OrderStore,
    OrderItemStore,
  },
  plugins: [
    createMutationsSharer({
      predicate: [
        'OrderItemStore/add',
        'OrderItemStore/reset',
        'OrderItemStore/set',
      ]
    })
  ],
});

export default store;
```

## vuex/index.js



Cornetto	2,50 €	Entfernen
Magnum	2,60 €	Entfernen
Calippo	1,80 €	Entfernen

**@hola\_soy\_milk**

Cornetto 2,50 €

Magnum 2,60 €

Calippo 1,80 €

**6,90 €**



And then what happened?

@hola\_soy\_milk



# Future additions

- Company Subsidies
  - Stability improvements
  - Printing on the same device
-

Ok so that was about 2 years ago, what happened since?



2 years later...  
sup?

- UI Improvements
-

Mitarbeiter, Mitarbeiter (001) ▼



Verfügbare Reserven

Getränke

Speisen

Sonstiges

Menü - Spezial

Menü - Warm Fle...

Menü - Warm Ve...

Salat 1 - Fleisch

Salat 2 - Vegetari...

Salat 3 - Thunfisch

Suppe zum Menü

Suppe zum Spezial

Salatbuffet

Suppe solo

Gebäck dunkel

Suppe zum Spezial

Salatbuffet groß

Menü - Schnitzel



1

Menü - Salat Fleisch



€ 4,80

**Gesamt**

**€ 4,80**

Bar

SumUp (M)

SumUp

Chipkarte

2 years later...  
sup?

- UI Improvements
  - Card payments
-

@hola\_soy\_milk



@hola\_soy\_milk



Signatures

**Payment Processing**

**Sign with online API**

**Receipt with QR code**

Bitte warten

€ 4,80 zu zahlen

Mitark



Menü - Salat Fleisch

enü - Spezial

Menü - Warm Fle...

Menü - Warm Ve...

```
def qr_code(string, qr_size=6)
  # Adapted from
  # https://stackoverflow.com/questions/23577702/
  # printing-qr-codes-through-an-esc-pos-thermal-printer
  s = string.size + 3
  lsb = (s % 256).chr
  msb = (s / 256).chr

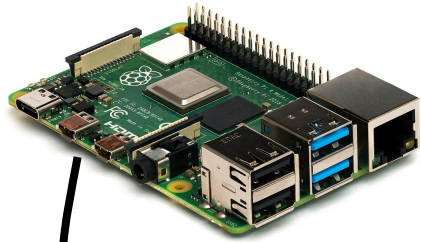
  # https://code.google.com/p/python-escpos/wiki/Usage
  escpos = ""
  escpos << "\x1D\x28\x6B\x03\x00\x31\x43#{qr_size.chr}"
  escpos << "\x1D\x28\x6B\x03\x00\x31\x45\x33"
  escpos << "\x1D\x28\x6B#{lsb}#{msb}\x31\x50\x30"
  escpos << string
  escpos << "\x1D\x28\x6B\x03\x00\x31\x51\x30"

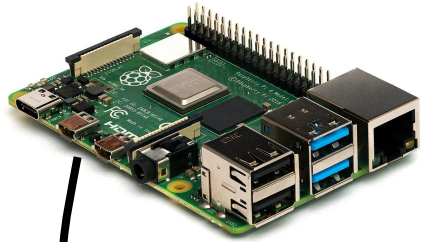
  # writing byte streams directly to the serial port
  printer.write escpos
end
```

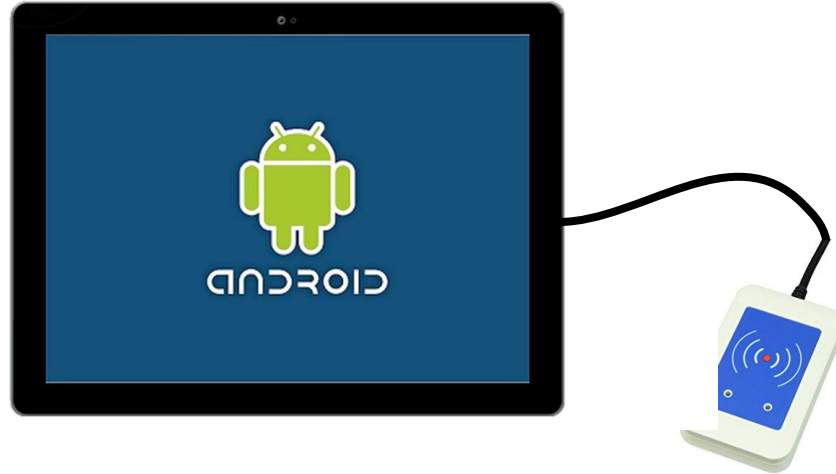


2 years later...  
sup?

- ~~UI Improvements~~
  - ~~Card payments~~
  - All-in-one Android app
-



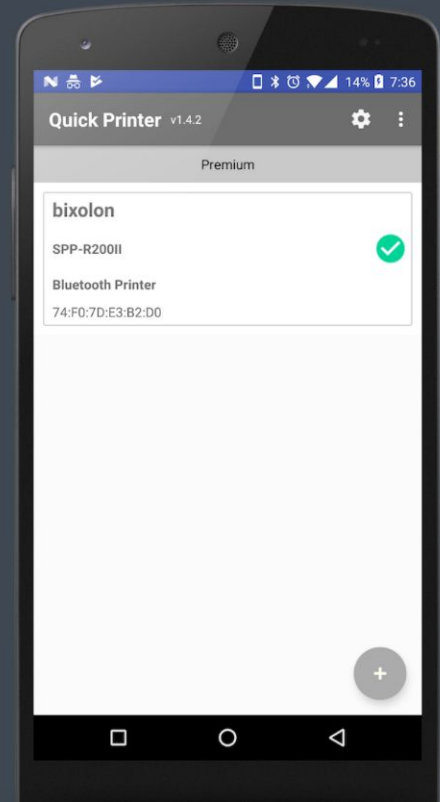




## App for Android

Quick printer is an Android application that allows you to add and configure receipt printers (POS printers) through different connection types

Download Quick Printer from Google Play [Here](#)



<RIGHT><SMALL>Datum: 2021/05/14 16:42:07<BR>

<LEFT><SMALL>Ramón Huidobro inc<BR>

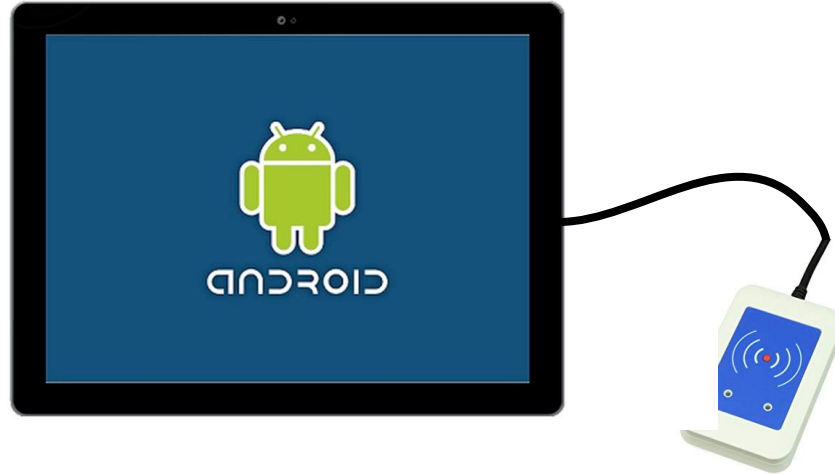
<BIG>Ramón Huidobro<BR><BR>

<BIG>Putenschnitzel<BR><BR>

<SMALL>Duplikat

<BR><BR><BR><BR><BR><CUT>

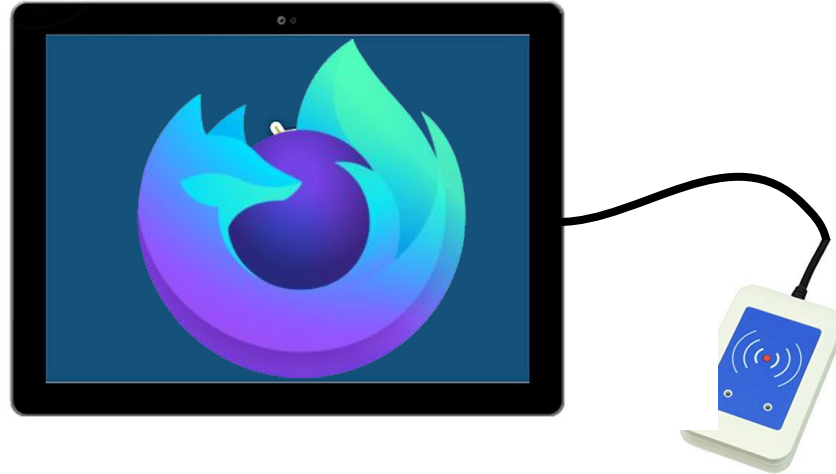
```
esc_data = render('point_of_sale/dashboards/receipts/receipt', receipt: @receipt)
text = url_encode(esc_data)
url = "intent://#{text}#Intent;scheme=quickprinter;package=pe.diegoveloper.printerserverapp;end;"
```





about:config

dom.disable\_open\_during\_load = false



I learned a lot! Gosh, did I ever!

Working with  
hardware becomes  
less daunting over  
time

Cost-effective  
solutions can flourish  
into new  
technologies.

Don't let yourself get stuck: Ask for help!

Play around with  
your existing tech!



---

Ramón Huidobro  
hola@ramonh.dev  
<https://ramonh.dev>