

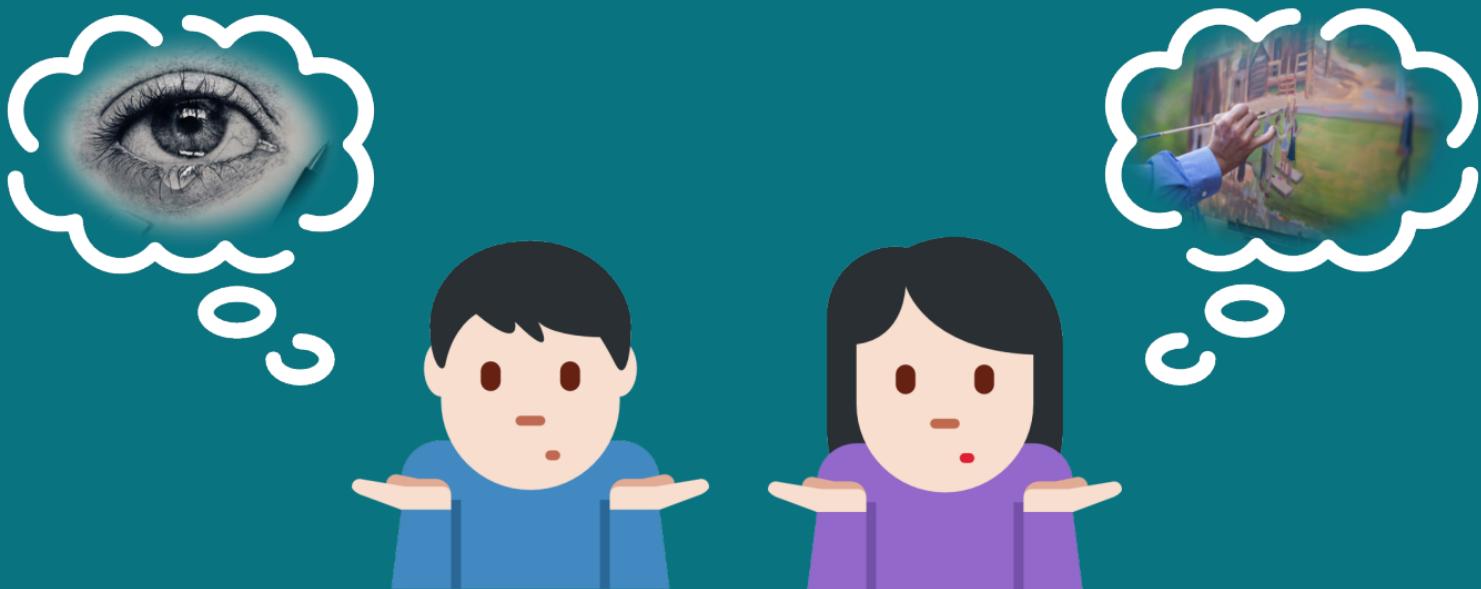


Visual Thinking in AXA

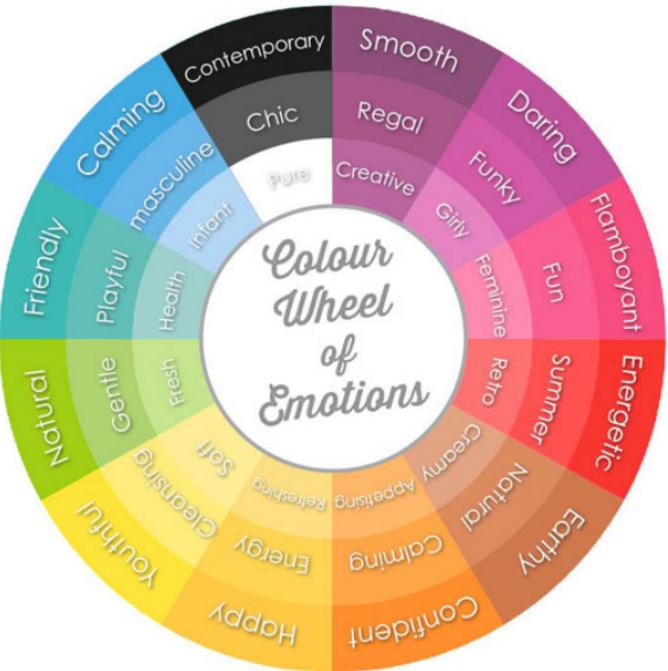
Virginia Armas Velásquez / Silvia de León González
November 2018 - Barcelona



Visual Thinking Introduction



Psychology of colors



IDEA



IDEA



PROBLEM



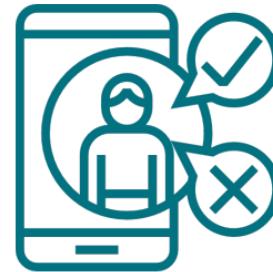
PROBLEM



OPINION



OPINION











Visual Thinking



Set of **tool and techniques** to represent visually the ideas, problems, tell histories and more!

Is not a skill for just a lucky few. **Everybody** is born with the ability to think in picture



75%-80%

of our brain is made to understand and process **images**

Impact of our work and teams



Communicate faster



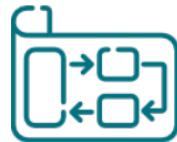
Message received



Speaking the same language



Feedback



Clarify process



Simplify the steps



Fun!



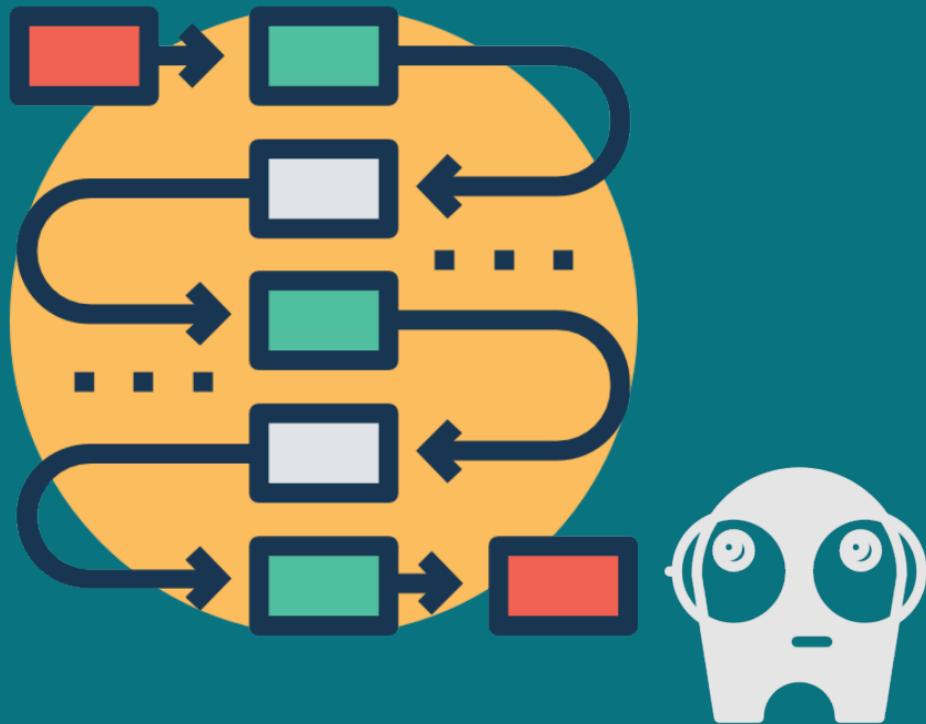
To the point



Hidden links



Process Visualization

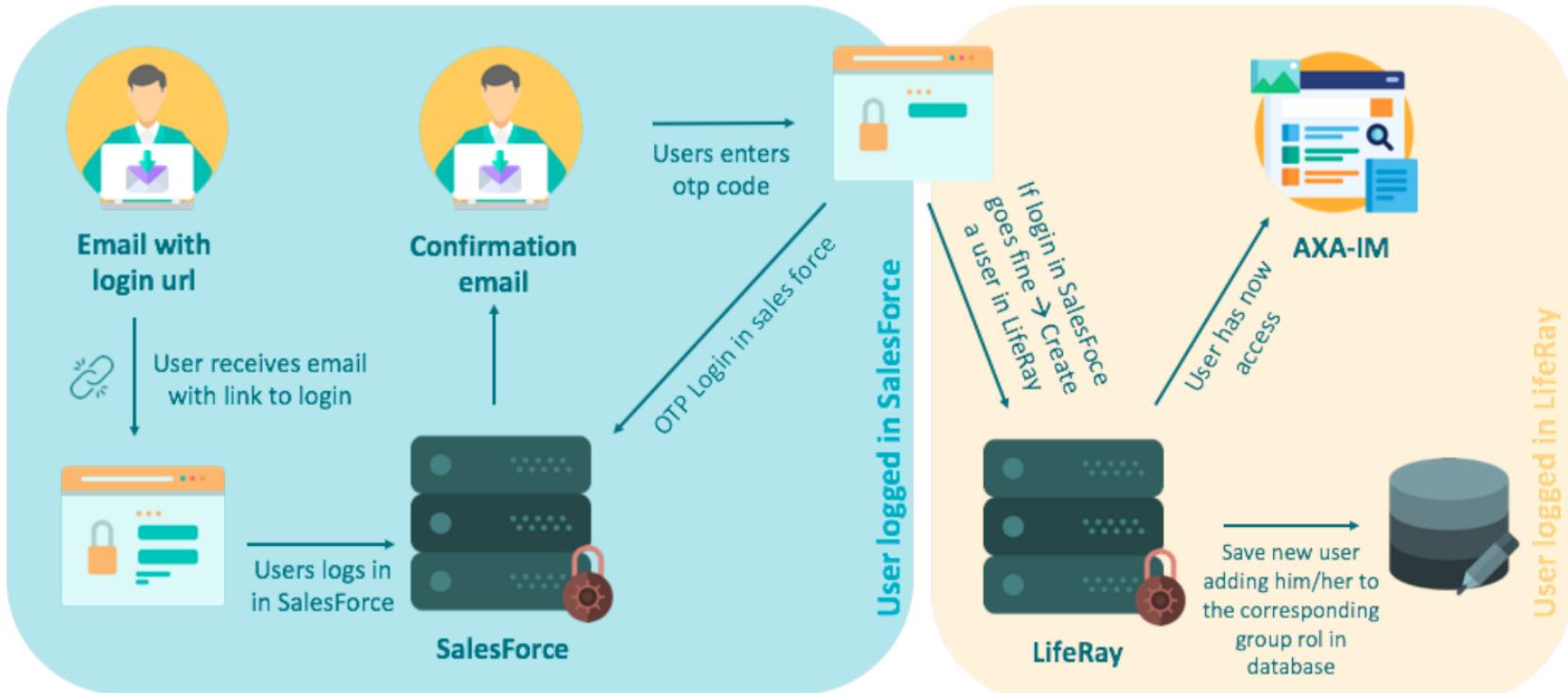


AIM: Login

To login, user receives an email which contains a link that redirects him to SalesForce site, there, he/she can insert user and password. Then, user receives a new email with a code to end the login process. In SalesForce site user inserts the code received, if it works, a user is created in LifeRay with his/her corresponding role. At this moment, user is finally logged in AIM and can see all its pages.



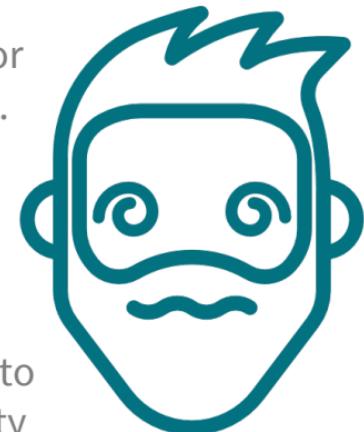
AIM: Login – Visual



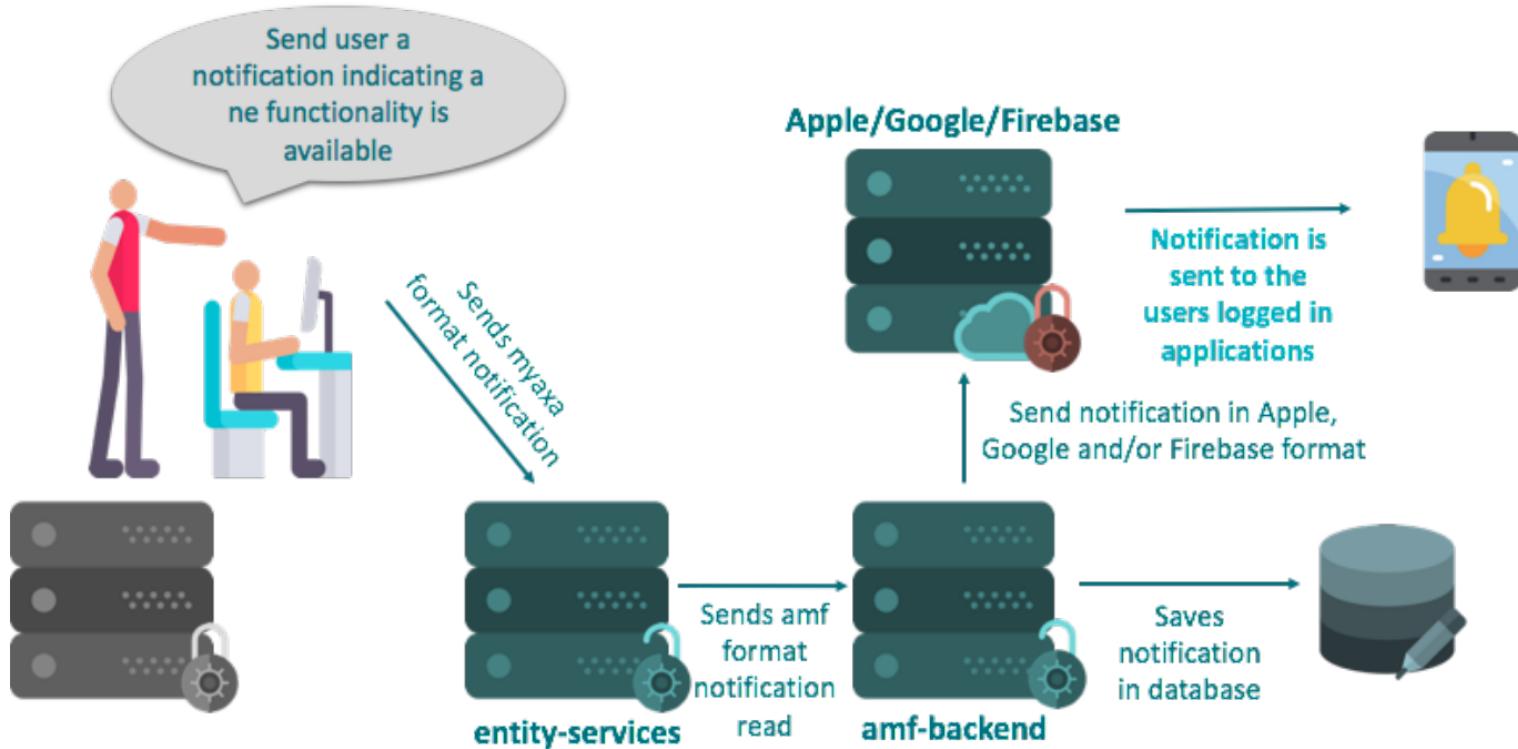
MyAXA: Push notifications

When entity wants to send a push notification to the users, they have to send it through the Entity Services, which will then send it to the AMF Backend and, finally, this server will convert it to the Apple, Google and/or Firebase format. The AMF Backend saves the notification in the database. Once Apple and Google receive the notification it sends it to all the users logged in applications.

When users read the notification. The MyAXA app request to the mobile-middleware to set that notification to read. Mobile middleware requests to Entity Services to set that notification to read (in myaxa format) and Entity services request it to AMF Backend in (the standard format). AMF Backend will update the database.

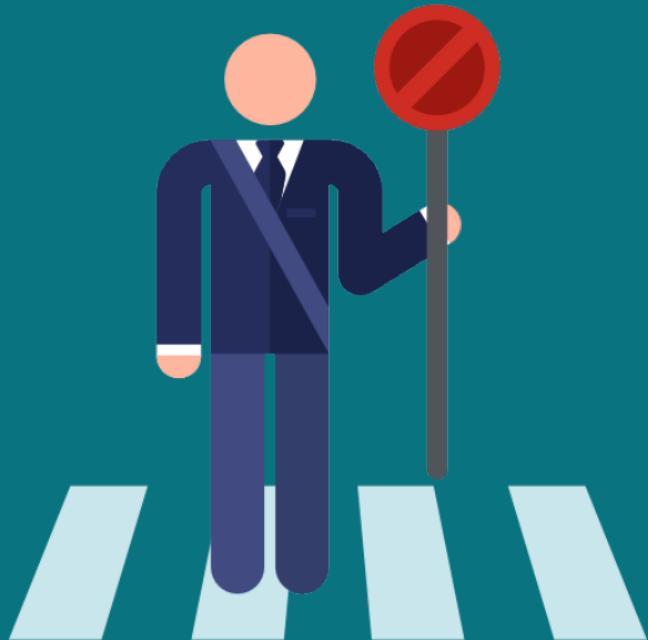


MyAXA: Push notifications – Visual





SAY NO to boring documentation



Readme files

Code guide style

- Eslint
- javascript airbnb styleguide
- Tslint
- typescript airbnb styleguide

More details about why we should lint our javascript code and our styleguide specs can be found [here](#).

Also, you can see the [AXA France Style Guide](#).

Commits conventions

To take advantage of the autogenerator of changelogs, [conventional-changelog](#), is need to follow the following conventional commits message format: <https://github.com/pvdlg/conventional-changelog-metahub>

```
commit -m "feat: the most awesome :shit: feature"
```

You can save time and memory using the next VSCode plugin: <https://github.com/KnisterPeter/vscode-committizen>

Branching strategy

Branching strategy is the way AXA Group Solutions Spain will use GIT for managing the lifecycle of the source code. Also the branching strategy is aligned with the CI platform tasks in order to manage release generation.

You can find the current branching strategy [here](#)



Code guide style



- ★ Eslint
- ★ javascript airbnb styleguide
- ★ Tslint
- ★ typescript airbnb styleguide

Want more? [here](#) --> More details about why we should lint our javascript code and our styleguide specs can be found [here](#).

Also, you can see the [AXA France Style Guide](#).

Commits conventions

To take advantage of the autogenerator of changelogs, [conventional-changelog](#), is need to follow the following conventional commits message format: <https://github.com/pvdlg/conventional-changelog-metahub>

```
commit -m "feat: the most awesome :shit: feature"
```

You can save time and memory using the next VSCode plugin: <https://github.com/KnisterPeter/vscode-committizen>

Branching strategy



Confluence pages

Javascript Development

Created by Unknown User (jvaldaura), last modified by Unknown User (beatriz.martin) on 02. May 2016

Content

- Development tools and workstation setup
- Coding Style Guide
- Architecture
- Useful libraries
 - Development frameworks
 - Testing frameworks
 - AXA libraries
 - JavaScript transpilers/compilers
- Needed Accounts
- Known issues in the frameworks
- Testing
- Other resources
- Links of interest

Development tools and workstation setup

- Install NODEJS: <https://nodejs.org/en/download/>
 - For windows see: HOW TO: Installing NodeJS and Strongloop in Windows and How to Execute npm install for
- Install NVM in order to be able to switch between NodeJS versions. <https://github.com/creationix/nvm>
- IDE
 - Any other tool is not allowed (for example Sublime text)
 - Webstorm: Javascript IDE Download
 - HOW TO ESLint Integration in Webstorm
 - Visual Studio Code: Alternative IDE Download
- SourceTree: Git client [Download](#)
- Configure eslint hook: [ESLint + Pre-commit hook](#)
- conEmu: Windows console emulator with tabs [Download](#)
- Configure NPM Private repository access: Using Artifactory as NPM private registry
- Test environment:
 - git clone <https://github.com/AXA-GROUP-SOLUTIONS/amf-commons-node.git>
 - cd amf-commons-node.git
 - npm install
 - If npm install works well your NPM is successfully installed.
- Configure ESLINT
 - Webstorm: [HOW TO ESLint Integration in Webstorm](#)
 - VisualStudio Code: TBD.



Javascript Development

Created by ARMAS Virginia, last modified just a moment ago

- Development tools and workstation setup
- Coding Style
- Architecture
- Useful libraries
- Needed Accounts
- Testing
- Other resources
- Links of interest

Development tools and workstation setup

1. NODEJS

<https://nodejs.org/en/download/>

① For windows see:

- HOW TO: Installing NodeJS and Strongloop in Windows
- How to Execute npm install from Private Github Repository through SSH in Windows 7

2. NVM

In order to be able to **switch between NodeJS versions** → <https://github.com/creationix/nvm>

3. IDE

⚠ Any other tool is not allowed (for example Sublime text)

Webstorm

Javascript IDE Download → HOW TO ESLint Integration in Webstorm

Visual Studio Code

① Alternative IDE Download

4. Sourcetree

Git client → [Download](#)

5. eslint

Configure **eslint hook** → [ESLint + Pre-commit hook](#)

Reports

Technologies

On frontend, the technology used is Angular 4.3.2, with the AXA template Angular (*deprecated*). On backend, the technology used is Loopback 3.0.0. The login system is PassAXA.

Environments

<https://qosmos.corp.intraxa>. For more information (dev environments, Sonar) please check this page: [Technical Documentation](#)

Features

Developed

Login/logout, dashboard, general report page, "My Report" page for IT Owners (see only their applications), export weekly data to Excel, mobile support, input smiley and comment, update smiley and comment report, annual percentage, average percentage of the week, percentage of completed report, add application, search applications, edit application, create user, list of users, edit user

Pending

Here some features not included on the scope of the project to implement in another phase. Wizard, management of automatic email for reminders of report, export members and applications, search for users, search of applications using attributes, Integration with Analytics

User stories developed: 36

Story points: 261

Sprints: 6

Work days: 55

	Sprint 1	Sprint 2	Sprint 3	Sprint 4
06 Nov 2017 - 17 Nov 2017	06 Nov 2017 - 17 Nov 2017	20 Nov 2017 - 01 Dec 2017	04 Dec 2017 - 06 Nov 2017	18 Dec 2017 - 08 Jan 2018
Compromised story points	50	86	103	112
Pending story points (from last sprint)	0	50	65	38
Removed from Sprint	-	-	-	-
Added to Sprint	-	-	-	8 (Added QBR-13 on 10/01)
Working Days	10	10	8 (2 holidays - December 6 and 8)	10
Total story points completed	0	21	49	112
Total story points incomplete	50	65	54	0



Technologies

Frontend

- Angular 4.3.2
- AXA template Angular (*deprecated*)

Backend

- Loopback 3.0.0

Login system

- PassAXA

Features Developed

- Login/Logout
- Dashboard
- General report page
- "My Report" page for IT Owners (see only their applications)
- Export weekly data to Excel
- Mobile support
- Input smiley and comment
- Update smiley and comment report
- Annual percentage
- Average percentage of the week
- Percentage of completed report
- Add application
- Search applications
- Edit application
- Create user
- List of users
- Edit user

Environments

<https://qosmos.corp.intraxa>

For more information (dev environments, Sonar) please check this page: [Technical Documentation](#)



	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6
06 Nov 2017 - 17 Nov 2017	06 Nov 2017 - 17 Nov 2017	20 Nov 2017 - 01 Dec 2017	04 Dec 2017 - 06 Nov 2017	18 Dec 2017 - 08 Jan 2018	22 Jan 2018 - 05 Feb 2018	06 Feb 2018 - 14 Feb 2018
Compromised story points	80	86	103	112	86	86
Pending story points (from last sprint)	0	50	65	38	-	49
Removed from Sprint	-	-	-	-	-	-
Added to Sprint	-	-	-	8 (Added QBR-13 on 10/01)	18 (Overspent on 23-Jan-2018)	8
Working Days	10	10	8 (2 holidays - December 6 and 8)	10	10	7
Total story points completed	0	21	49	112	15	64
Total story points incomplete	50	65	54	0	49	0

Features to Develop

Here some features not included on the scope of the project to implement in another phase.

- ★ Wizard
- ★ Management of automatic email for reminders of report
- ★ Export members and applications
- ★ Search for users
- ★ Search of applications using attributes
- ★ Integration with Analytics



Templates you will want to use



Trello – During the meeting

What is the problem !?

- I'm too late for work (after 10 am)

+ Add another card

Things I've tried 🤔

- Get a shower the night before
- Setup more than one alarm
- Leave earlier with the car (too much time looking for parking)

+ Add another card

New Ideas! 💡

- Go to a coffee with less people
- Take breakfast to the office
- Use the bus/subway

+ Add another card



I'm too late for work
(after 10 am)

Problem



- Get a shower the night before
- Setup more than one alarm
- Leave earlier with the car (too much time looking for parking)

Things I've tried



- Go to a coffee with less people
- Take breakfast to the office
- Use the bus/subway

New ideas

Trello – During the meeting

The image shows a Trello board with the following structure:

- Goal**: Get a new washing machine
+ Add another card
- What do I need to DO?**: Save money, Look for sales, Decide the type of washing machine
+ Add another card
- Who do I NEED?**: Plumber for installation, Partner, Friends (for recommendations)
+ Add another card
- Next steps**: Write on whatsapp group to ask for recommendation, Go with my partner to stores to analyse options
+ Add another card
- Checkpoint**: 1 week
+ Add another card

Goal

Get a new washing machine



What do I need to do

Decide the type of washing machine

Save money

Look for sales

Friends
(for recommendations)

Who do I need?



Plumber - for installation



Partner – decision together



Next Steps



Write on whatsapp group to ask for recommendation

Go with my partner to stores to analyze options



Checkpoint

Let's refactor!

Goal

Remove duplicated code



Good Practices

- ✓ Work on other branch
- ✓ Adapt tests to new code
- ✓ Commits descriptions
- ✓ (not “Refactor” and push!)



Warning

“special” code
Sensible functions
Update documentation

I need help!



Frontend



Architect



References

- Visual Thinking. Willemien Brand
- Psychology of colors
 - <https://www.medicaldaily.com/color-psychology-10-ways-colors-trick-you-every-day-295460>
 - http://www.arttherapyblog.com/online/color-psychology-psychologica-effects-of-colors/#.W-F_Y3pKjOR
- Images
 - <http://www.freepik.com>
 - <https://www.flaticon.com/authors/smashicons>

