

UCI

Frontend training

Agenda

1. Pre-Requirement
2. Novicell Frontend package
3. Atom design
4. Fractal
5. Handlebar
6. Styles in Fractal
7. PostCSS & BEM
8. More style trick and good practices
9. Scripts in Fractal
10. Development process and deployment
11. Integration with Sitecore
12. ECOWEB case overview
13. QA

Pre-Requirement

Pre-Requirement

To attend the workshop you need good Knowledge on:

- CSS
- HTML5
- Javascript

Basic notion on:

- Git
- NPM
- CSS preprocessor
- Web development process

Pre-Requirement

Before the workshop, you need to have your local computer setup with:

- Git installed
- Optionally some git client (ex: gitkraken)
- Node.js min. v. 14.15.0 with NPM 6.14.8 installed
- Visual Code Studio installed
- Novicell frontend framework installed in a local folder. Use the following repository on master branch for it:
<https://github.com/agiraud/frontend-training>

Novicell frontend package

framework that speeds up front-end development process
and ensure frontend code quality and optimisation.

Novicell Frontend package technologies

- <https://docs.npmjs.com/about-npm>: software registry
- <https://webpack.js.org/>: assets bundle
- <https://postcss.org/>: css processor
- <https://stylelint.io/>: css style convention
- <https://eslint.org/>: JavaScript quality
- <https://babeljs.io/>: JavaScript compiler
- <https://fractal.build/>: component libraries & style guides
- <https://handlebarsjs.com/>: HTML templating

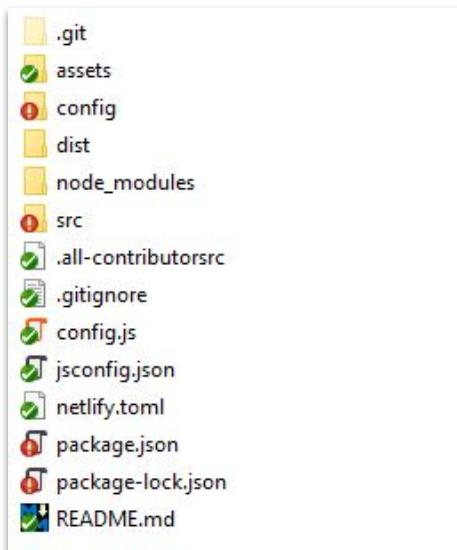
Novicell Frontend package tools

- Pre-configured libraries
- Set of NPM Scripts
- Default atom folder structure
- Component helper
- Default Grid and variable list
- SVG Sprites
- Open-source project

<https://github.com/Novicell/novicell-frontend>

Novicell Frontend Setup

<https://github.com/Novicell/novicell-frontend#setup>



A screenshot of a Windows Command Prompt window titled 'npm'. The command 'npm run build:dev' is being executed. The output shows the build process for sprites, images, styles, and fonts using tools like node, cross-env-shell, and postcss.

```
Microsoft Windows [Version 10.0.18362.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

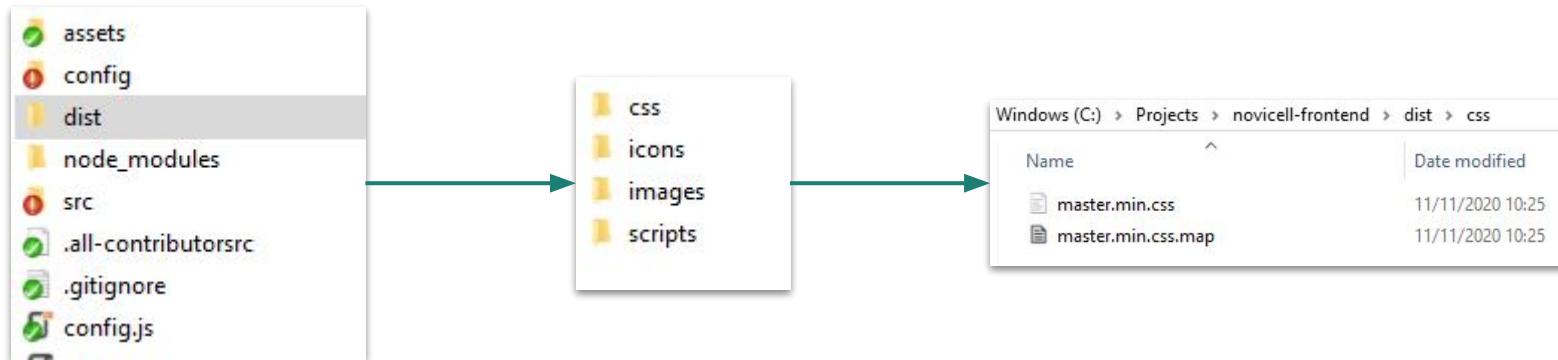
C:\Projects\novicell-frontend>npm run build:dev

> novicell-frontend@5.0.0 build:dev C:\Projects\novicell-frontend
> cross-env NODE_ENV=development npm-run-all -l sprites images fonts test webpack

[sprites] > novicell-frontend@5.0.0 sprites C:\Projects\novicell-frontend
[sprites] > cross-env-shell node config.sprites(sprite.js -i 'assets/icons/**/*.*.svg' -o $npm_package_config_DIST/icons/icons.svg
[sprites]
[images] > novicell-frontend@5.0.0 images C:\Projects\novicell-frontend
[images] > node config/images/imagemin.js
[images] minified: C:\Projects\novicell-frontend\dist\images\favicons-master.png
[styles] > novicell-frontend@5.0.0 styles C:\Projects\novicell-frontend
[styles] > cross-env-shell postcss $npm_package_config_CSS_MODULES --dir $npm_package_config_DIST/css --ext min.css --config config/styles/postcss.config.js
[styles] Browserslist: caniuse-lite is outdated. Please run next command `npm update caniuse-lite browserslist`
```

Exercice 1 - 5 mins

1. From Visual Studio Code, open the project with “Open Folder”
2. Open a new terminal
3. Try to run “npm run build:dev”
4. Check that the dist folder has been created and master.min.css compiled



Atomic design

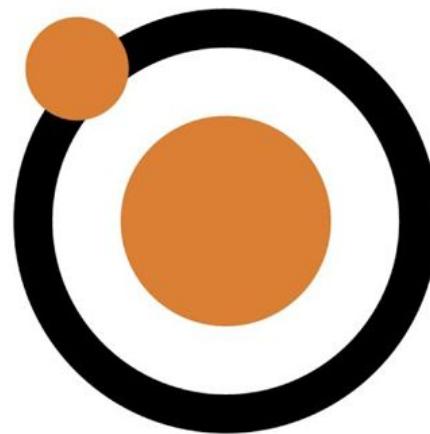
“We’re not designing pages, we’re designing systems of components” — Stephen Hay

atomic design

Must read:

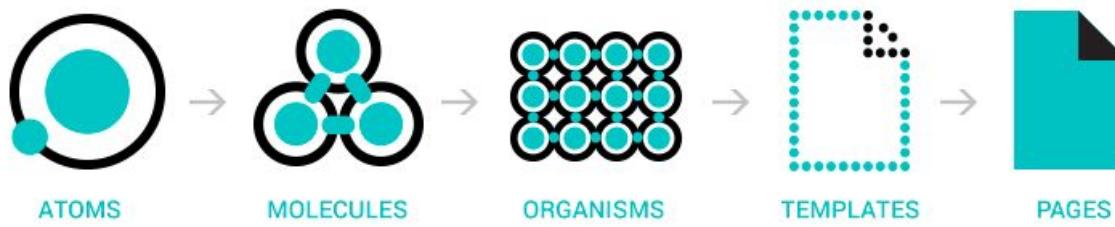
atomic design by Brad Frost

<https://atomicdesign.bradfrost.com/>



ATOMIC DESIGN

atomic design

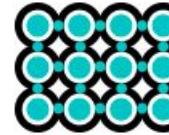




ATOMS



MOLECULES



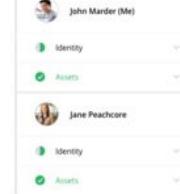
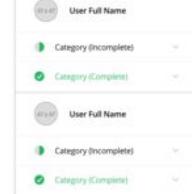
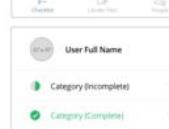
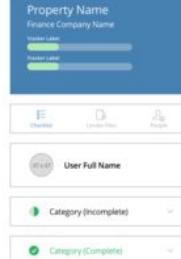
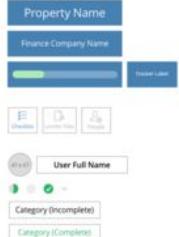
ORGANISMS



TEMPLATES



PAGES



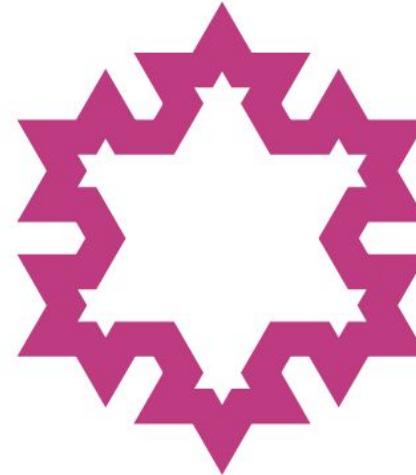
Fractal

Fractal is a tool to help you build and document web component libraries, and then integrate them into your projects.

Fractal

- Component libraries framework
- Style Guides platform
- Template & Data-driven with Handlebars
- Atom design as organisational model

<https://fractal.build/guide/>



Run fractal

Run fractal as local server:
`npm run fractal`

```
C:\Projects\novicell-frontend [master ≡ +1 ~4 -0 !]> npm run fractal
> novicell-frontend@5.0.0 fractal C:\Projects\novicell-frontend
> fractal start --sync

Deprecated configuration file name! Rename fractal.js to fractal.config.js.

Fractal web UI server is running!
Local URL: http://localhost:3000
Network URL: http://192.168.1.172:3000
BrowserSync UI: http://localhost:3002

Use ^C to stop the server.
```

The screenshot shows a browser window titled "Component Library" at the URL "localhost:3000". The interface includes a search bar, a sidebar with navigation links like "COMPONENTS", "DOCUMENTATION", and "INFORMATION", and a main content area with a heading "Welcome to your component library". Below this, there's a section titled "Component statuses" with a table showing three rows: "PROTOTYPE" (Description: Do not implement.), "WIP" (Description: Work in progress. Implement with caution.), and "READY" (Description: Ready to implement.).

Label	Description
PROTOTYPE	Do not implement.
WIP	Work in progress. Implement with caution.
READY	Ready to implement.

Fractal overview

Current component

The screenshot illustrates the Fractal Component Library interface. On the left, a sidebar titled "Component Library" shows a tree structure of components under "COMPONENTS". The "Examples" section contains "Atom Example", "Molecule Example", and "Organism Example", with "Organism Example" currently selected. Other sections include "Atoms", "Docs", "DOCUMENTATION", and "Index". The bottom of the sidebar displays "INFORMATION" and "Built on: 11/11/2020".

The main content area is titled "Organism Example" and features two "Molecule" components. The first molecule has a heading "Molecule 1 heading" and the text "The text in the @molecule-example." The second molecule has a heading "Molecule 2 heading" and the text "The text in the @molecule-example." A "WIP" badge is visible in the top right corner of the main area.

Annotations on the left side point to the sidebar with the text "Component organization". Annotations on the right side point to the main content area with the text "Component definition". A large green arrow points from the "Organism Example" title towards the "Molecule 2 heading".

Below the main content, a table provides details for the selected component:

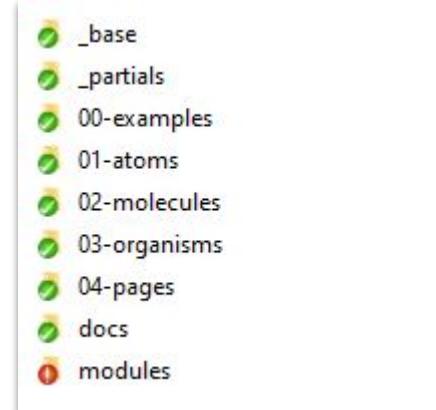
HTML	View	Context	Info	Notes
<pre><div class="container"> <div class="row"> <div class="col-xs-12 col-sm-6"> {{> '@molecule-example' moleculeData1 }} </div> <div class="col-xs-12 col-sm-6"> {{> '@molecule-example' moleculeData2 }} </div> </div> </div></pre>	View			

Fractal, component definition



Fractal, folder structure

- _base
- _partials
- 01-atoms
 - component-a
 - component-b
- 02-molecules
- 03-organisms
- 04-pages
- docs
- modules



Fractal component

individual pieces of your website's UI

└─ components

 └─ components.config.json



 └─ components.hbs (Handlebar)



 └─ components.js

 └─ README.md

 └─ components.css (postCSS)

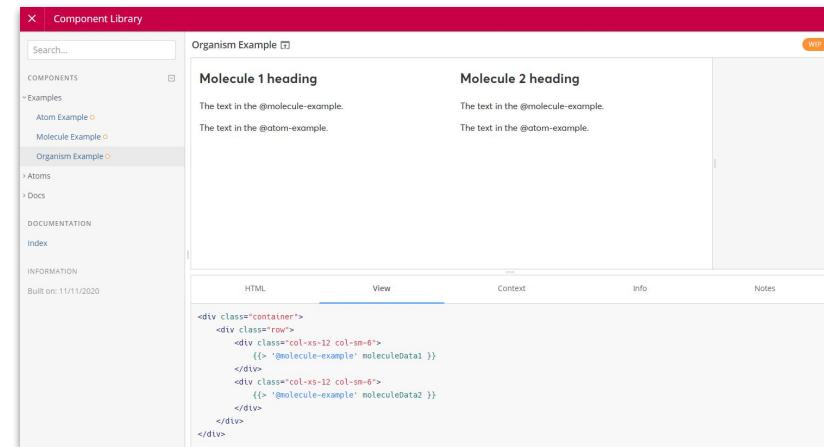
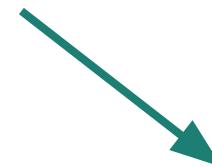
```
/*  
 * Atom Example  
 */  
  
:root {  
  --atom-example__text-color: var(--color-plum);  
}  
  
.atom-example {  
  color: var(--atom-example__text-color);  
}
```

<https://fractal.build/guide/components/>

<https://github.com/Novicell/novicell-frontend/wiki/Fractal-guidelines>

Exercice 2 - 5 mins

1. Run fractal and check the fractal portal structure



The screenshot shows the Component Library interface with the 'Organism Example' component selected. The left sidebar lists components like 'Atom Example', 'Molecule Example', and 'Organism Example'. The main area displays two sections: 'Molecule 1 heading' and 'Molecule 2 heading', each containing placeholder text. Below these sections is an 'HTML' tab, which is currently active, showing the underlying React component code:

```
<div class="container">
  <div class="row">
    <div class="col-xs-12 col-sm-6">
      {(> '@molecule-example' moleculeData1 )}
    </div>
    <div class="col-xs-12 col-sm-6">
      {(> '@molecule-example' moleculeData2 )}
    </div>
  </div>
</div>
```

Handlebar

Fractal, Handlebar



Handlebars is a simple templating language.

- [Simple Data binding](#)
- [Nested input objects](#)
- [Evaluation context](#)
- [Partial](#)
- ...

```
{  
  "title": "Molecule Example",  
  "status": "wip",  
  "context": {  
    "heading": "Molecule example heading",  
    "text": "Solid: We need a similar pattern, but want to reuse @atom-example",  
    "atomExample": {  
      "text": "Dry: We can reuse @atom-example and don't have to copy/paste it."  
    }  
  }  
}
```



```
<div class="molecule-example">  
  <h2 class="molecule-example_heading">{{ heading }}</h2>  
  <div class="molecule-example_solid-pattern">  
    <p class="molecule-example_text">{{ text }}</p>  
  </div>  
  <div class="molecule-example_dry-pattern">  
    {{> '@atom-example' atomExample }}  
  </div>  
</div>
```

Fractal, Handlebar



Helper:

- if
- unless
- each
- with
- ...

```
<ul class="people_list">
  {{#each people}}
    <li>{{this}}</li>
  {{/each}}
</ul>
```

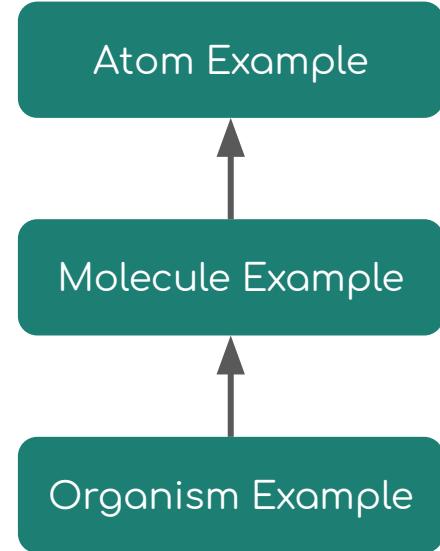
```
{
  people: [
    "Yehuda Katz",
    "Alan Johnson",
    "Charles Jolley",
  ],
}
```

```
<ul class="people_list">
  <li>Yehuda Katz</li>
  <li>Alan Johnson</li>
  <li>Charles Jolley</li>
</ul>
```

Fractal, Handlebar sub-components



- Including Sub-components
`{> '@molecule-example' }`
- Sub-components with context data
`{> '@molecule-example' moleculeData1 }`



Fractal, generate component

- `createComponent -t p -n myComponent`

- a: atom
- m: molecule
- o: organisme
- p: page

```
$.createComponent -t p -n myAwesomePage
Created folder in D:\webdev\Novicell-frontend\src\04-pa...
myAwesomePage.css was created
myAwesomePage.hbs was created
myAwesomePage.json was created
```

- Note: run “npm link” first

Exercice 3 - 20 mins

1. Create two atom components:
 - a. "title" (as a `<h2>` element)
 - b. "summary" (as a `<p>` element)
 - c. for both component set a text as context



Placeholder text for the footer area.

Consectetur adipiscing

2. Create a molecule component “card”:
 - a. With an image, a atom title, and atom molecule
 - b. Set a context for the image source, title text and summary text



Consectetur adipisci

orem ipsum dolor sit amet, consectetur

3. Create an organism component “last-post” which lists “card” components:
 - a. Create a context object with 3 cards, use the #each instruction to list them



Exercice 3 - 15 mins - anexe

1. "title" html result



```
<h2 class="title">Consectetur adipiscing</h2>
```

2. "summary" html result



```
<p class="summary">Lorem ipsum dolor sit amet, consectetur adipiscing  
elit, sed do eiusmod tempor incididunt</p>
```

3. "card" html result



```
<div class="card">  
    
  <div class="card__content">  
    <h2 class="title">Consectetur adipiscing</h2>  
    <p class="summary">Lorem ipsum dolor sit amet,  
      adipiscing elit, sed do eiusmod tempor incididunt</p>  
  </div>  
</div>
```

4. "post-list" html result



```
<div class="last-post">  
  <ul class="last-post__list">  
    <li class="last-post__list__item">  
      <div class="card">  
          
        <div class="card__content">  
          <h2 class="title">Consectetur adipiscing</h2>  
          <p class="summary">Lorem ipsum dolor sit amet...</p>  
        </div>  
      </div>  
    </li>  
    ...  
  </ul>  
</div>
```

Styles in Fractal

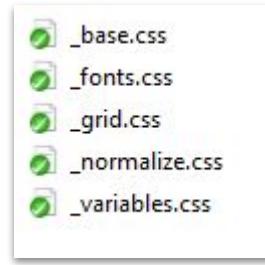
Fractal, Add CSS files

- Generated with `createComponent`
- Must be imported into `master.css` or into a master css file into the `src/Modules` directory

```
@import '../04-pages/myAwesomePage/myAwesomePage.css';  
  
src/04-pages/myAwesomePage/myAwesomePage.css  
  
.box {  
    margin: 50px auto;  
    width: 100px;  
    height: 100px;  
    background-color: blue;  
}
```

Fractal, style base

- _base
 - _base.css
 - _fonts.css
 - _grid.css
 - _normalize.css
 - _variables.css
- _partials
- ...



Fractal, npm commands

- npm run ...
 - build:dev
 - build:prod
 - fractal
 - fractal:build
 - styles
 - watch:styles
 - ...

```
C:\Projects\novicell-frontend [master = +1 ~4 -0 !]> npm run watch:styles

> novicell-frontend@5.0.0 watch:styles C:\Projects\novicell-frontend
> cross-env-shell postcss $npm_package_config_CSS_MODULES --dir $npm_package_config_DIST/css --ext min.css
  -config config/styles/postcss.config.js --watch --verbose
  --config config/styles/postcss.config.js --watch --verbose

> cross-env-shell postcss $npm_package_config_CSS_MODULES --dir $npm_package_config_DIST/css --ext min.css
  --config config/styles/postcss.config.js --watch --verbose

Processing src\modules\master.css...
```

Exercice 4 - 15 mins

1. Set style for the title atom



```
.title {  
    font-size: 1.6em;  
    text-transform: uppercase;  
    color: var(--color-primary);  
}
```

2. Set style for the card molecule



```
.last-post ul {  
    list-style: none;  
    display: inline-block;  
}  
  
.last-post ul li {  
    display: inline-block;  
    margin: 0 15px;  
}
```

3. Set style for the last-post module



```
.card {  
    height: 25em;  
    width: 20em;  
    overflow: hidden;  
    border-radius: 16px;  
    position: relative;  
    display: -webkit-box;  
    display: -ms-flexbox;  
    display: flex;  
    -webkit-box-orient: vertical;  
    -webkit-box-direction: normal;  
    -ms-flex-direction: column;  
    flex-direction: column;  
    -webkit-box-shadow: 15px 15px 27px var(--color-gray-light), -15px -15px 27px var(--color-white);  
    box-shadow: 15px 15px 27px var(--color-gray-light), -15px -15px -27px var(--color-white);  
}  
  
.card_content {  
    padding: 15px;  
}
```

4. Check the result for the 3 modules



CONSECTETUR
ADIPISCING

Consectetur
adipiscing elit,
sed do eiusmod
tempor incididunt



CONSECTETUR
ADIPISCING

Consectetur
adipiscing elit,
sed do eiusmod
tempor incididunt



CONSECTETUR
ADIPISCING

Consectetur
adipiscing elit,
sed do eiusmod
tempor incididunt

PostCSS & BEM

PostCSS aims to reinvent CSS with an ecosystem of custom plugins and tools

PostCSS

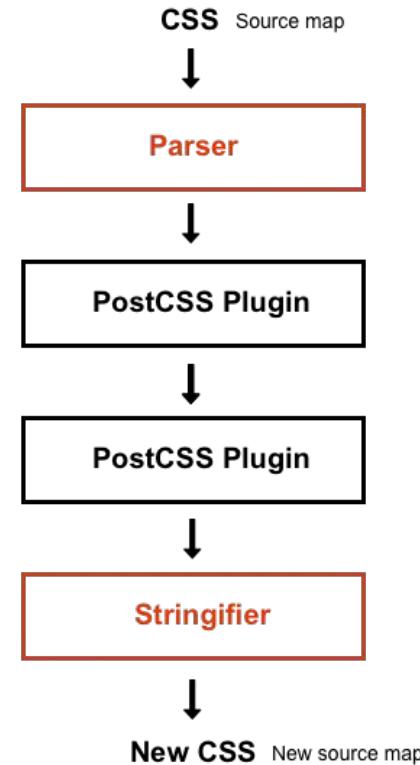
- PostCSS is a tool for transforming styles with JS plugins
- Working with the same principles of preprocessors such as Sass and LESS
- Ecosystem of custom plugins and tools



<https://github.com/postcss/postcss>

PostCSS

- Parses CSS into an abstract syntax tree ([AST](#))
- passes that AST through any number of “plugin” functions
- passes that AST through any number of “plugin” functions



<https://github.com/postcss/postcss>

PostCSS

.sass, .scss, .styl or .less ? nop,
just traditional CSS...

postcss allow to pick up and
configure how we really want
to work

```
.addresses {  
  &--title {  
    margin-bottom: 20px;  
  
    @media (--viewport-ms-min) {  
      margin-bottom: 30px;  
    }  
  
    @media (--viewport-sm-min) {  
      margin-bottom: 40px;  
    }  
  }  
  
  &--offices {  
    @media (--viewport-sm-min) {  
      margin-bottom: -40px;  
    }  
  
    > col {  
      &:last-child {  
        .addresses--office {  
          @media (--viewport-ms-max) {  
            margin-bottom: 0;  
          }  
        }  
      }  
    }  
  }  
}
```

PostCSS - Plugin

- [postcss-import](#)
transform @import rules by inlining content
- [postcss-preset-env](#)
convert modern CSS into something most browsers can understand.
This this the legacy cssnext, see [feature here](#)
- [postcss-nested](#)
unwrap nested rules like how Sass does it
- [cssnano](#)
optimise css code
- [postcss-reporter](#)
Report postcss plugin operations

<https://github.com/postcss/postcss>

BEM: Block Element Modifier

Methodology to create reusable and clear components in front-end development

- Easy
- Modular
- Flexible



BEM: Block Element Modifier

Main naming concepts

- **Block**, Encapsulates a standalone entity that is meaningful on its own.

```
<div class="block">...</div>
```

```
.block { color: #042; }
```

- **Element**, Parts of a block and have no standalone meaning.

```
<div class="block">  
    ...  
    <span class="block__elem"></span>  
</div>
```

```
.block__elem { color: #042; }
```

- **Modifier**, Flags on blocks or elements. Use them to change appearance, behavior or state.

```
<div class="block block--mod">...</div>  
<div class="block block--size-big  
      block--shadow-yes">...</div>
```

```
.block--hidden { }
```

```
.block__elem--mod { }
```

Result of all of this

```
.addresses {  
  &__title {  
    margin-bottom: 20px;  
  
    @media (--viewport-ms-min) {  
      margin-bottom: 30px;  
    }  
  
    @media (--viewport-sm-min) {  
      margin-bottom: 40px;  
    }  
  }  
  
  &__offices {  
    @media (--viewport-sm-min) {  
      margin-bottom: -40px;  
    }  
  }  
  
  > col {  
    &:last-child {  
      .addresses__office {  
        @media (--viewport-ms-max) {  
          margin-bottom: 0;  
        }  
      }  
    }  
  }  
}
```

BEM

postcss-nested

postcss-preset-env

Exercice 5 - 15 mins

1. Refactor card.css and last-post.css following the BEM rules and using nested style definition.
2. In card.css, create an alternative card version “highlight”, and apply this style to the first last-post component’s card.
Trick: use handlebar “if” option to apply the highlight alternative style to the first card



More style trick and good practices

Grid system

Novicell Frontend framework comes with a out of the box grid system

- 12 columns
- 5 breaking point: lg, md, sm, ms, xs
- Configurable
- \src_\base_\grid.css

```
<div class="container">
  <div class="row">
    <div class="col-xs-12 col-ms-6 col-sm-4 col-lg-4">
      <div class="row">
        <div class="col-xs-3 col-ms-3 col-lg-3">
          <div class="grid-example-col-content">
            <p>1st column</p>
          </div>
        </div>
        <div class="col-xs-3 col-ms-3 col-lg-3">
          <div class="grid-example-col-content">
            <p>2nd column</p>
          </div>
        </div>
      </div>
    </div>
```

1st column	2nd column	3rd column	4th column	5th column	6th column	7th column	8th column	9th column	10th column	11th column	12th column
------------	------------	------------	------------	------------	------------	------------	------------	------------	-------------	-------------	-------------

CSS style guideline

Must read:

[https://github.com/Novicell/novicell-frontend/wiki/CSS-\(PostCSS\)](https://github.com/Novicell/novicell-frontend/wiki/CSS-(PostCSS))



stylelint.io extension

A mighty, modern linter that helps you avoid errors and enforce conventions in your styles.

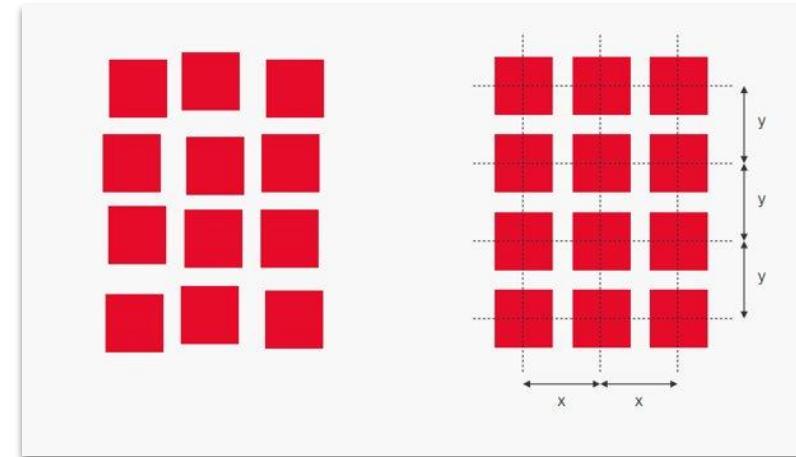
<https://stylelint.io/>

Check config in config/styles



Be pixel perfect

- A pixel-level design affinity in the encoding of HTML files and CSS styles is essential for pages to convey the stability, emotions, and concepts brought in during the design phase.
- To ensure an optimal level of quality, we can perform automated visual comparison tests with ghostinspector.com to detect any unexpected changes in graphical level.



Think mobile “first”

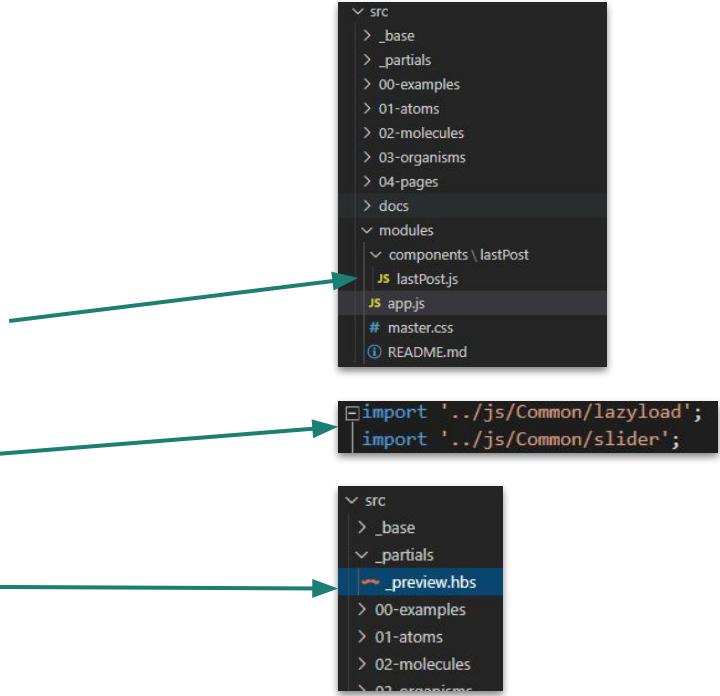
- Ensures an optimal user experience on mobile devices
- Favors, due to the nature of small devices, graphic creativity
- Optimize development as it is easier to scale to higher resolutions
- Ensures that content and interactions are reproduced optimally on any device
- Optimize page loading speed.



Scripts in Fractal

Add JS files

- Should be always added into the src/Modules/ directory
- Create a new folder for each component to maintain a good separate the concerns
- Add references to JS files into src/Modules/app.js
- Main references are included into the default layout file
- Compile script with
 - npm run webpack
 - npm run watch:webpack



ESLint extension

Find and fix problems in your
JavaScript code

<https://eslint.org/>

Check config in config/script



Babel extension

JavaScript compiler, Use next generation JavaScript, today

<https://babeljs.io/>



Put in next-gen JavaScript	Get browser-compatible JavaScript out
<pre>element.index ?? -1;</pre>	<pre>var _element\$index; (_element\$index = element.index) != null ? _element\$index : -1;</pre>

Exercice 6 - 5 mins

1. Add a js file lastPost.js into the project
2. Make sure that the js is executed in Fractal

```
console.info('Do something here');
```

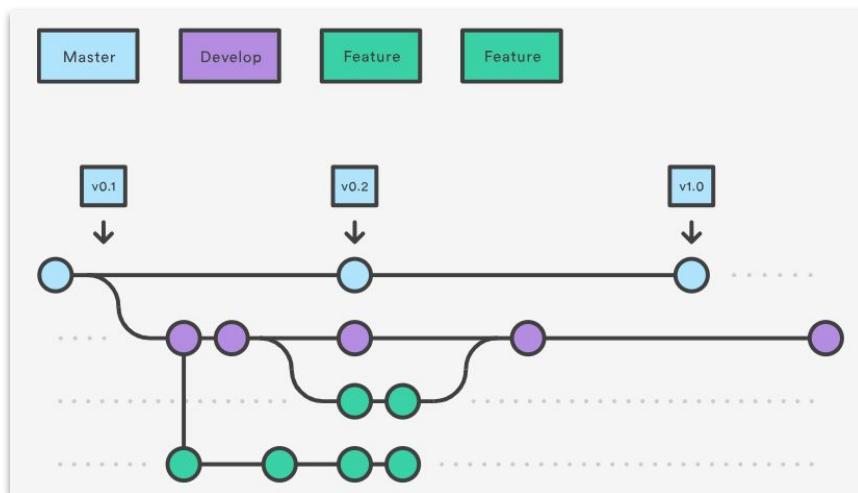


Development process and deployment

GitFlow

Standard branching model strategy

- Always create a new feature branch from develop for each User Stories
- Create sub-branches from features branches for each task and commit your changes there
- Merge your feature branches in develop by pull request
- Merge develop branch into master (or create release branches), tags each releases in Master



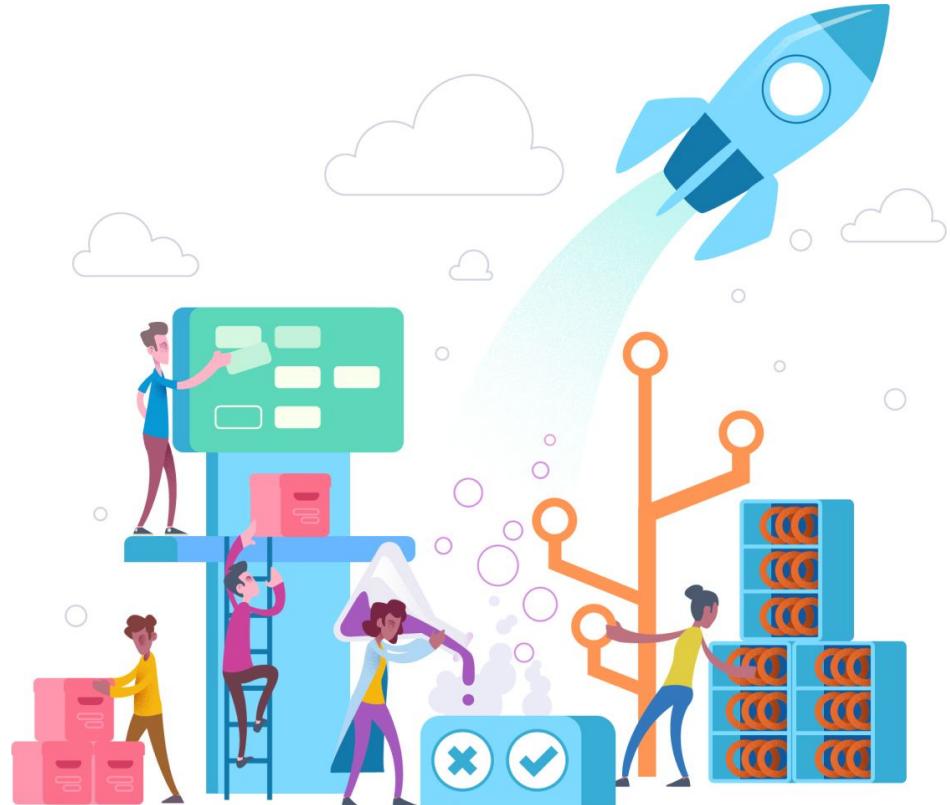
<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

https://dev.azure.com/Uci-EcowebDevOps/ecoweb-app/_wiki/wikis/ecoweb-app.wiki/16/Project-Development-Process

DevOps

Delivery application and services

- Code repository
- Project management
- Building pipeline
- Deploy process



<https://dev.azure.com/Uci-EcowebDevops/ecoweb-app/>

Code repository

- Git repository
- Same repository for back and front

The screenshot displays a software interface for managing a code repository. On the left, a sidebar provides navigation links for various repository features: Overview, Boards, Repos, Files, Commits, Pushes, Branches, Tags, Pull requests, Pipelines, Test Plans, and Artifacts. The 'Files' link is currently selected, indicated by a grey background.

The main content area is divided into two sections. On the left, a tree view shows the directory structure of the 'ecoweb-app' repository, including 'data', 'extrafiles', 'Feature', 'Foundation', 'License', 'Project', 'Serialization', and 'startup'. Below this tree view is a list of files: '.editorconfig', '.env.sample', '.gitignore', 'build-and-deploy-configurati...', 'docker-compose.reduce.yml', and 'docker-compose.yml'. On the right, a detailed file list is shown for the 'feature/ISSUE-35-TinTaeRefactoring' branch. This list includes columns for Name, Last change, and Commits. The files listed are 'data', 'extrafiles', 'Feature', 'Foundation', 'License', 'Project', 'Serialization', 'startup', '.editorconfig', '.env.sample', and '.env.sample'. The commit history for each file is also visible, such as '0246139d' for 'data' on Jan 7.

Name ↑	Last change	Commits
data	Jan 7	0246139d
extrafiles	Oct 28	1a29afa3
Feature	Nov 4	b00c97af
Foundation	Oct 28	1a29afa3
License	Jan 7	4e40ff7a
Project	Oct 29	054eeb99
Serialization	Oct 28	1a29afa3
startup	Apr 23	d2b73f0f
.editorconfig	Jun 2	cc70f1bd
.env.sample	Jan 20	839e9a05

Merging by PR process

PR: Pull request

- Testing and better stability
- Clearer responsibility
- Sharing knowledge
- Avoid conflict
- Meaningful git history



Merging by PR process

- Merge any feature to develop by PR
- Merge develop to master by PR
- Always at least one reviewer

The screenshot shows the 'Pull requests' page for the 'ecoweb-app' repository. The left sidebar includes 'Overview', 'Boards', 'Repos', 'Files', 'Commits', 'Pushes', 'Branches', 'Tags', 'Pull requests' (which is selected), 'Pipelines', 'Test Plans', and 'Artifacts'. The main area displays a message: 'Currently, no pull requests need your attention'. It includes a small icon of three people reviewing code and a link to learn more about pull requests.

The screenshot shows the 'New pull request' dialog for the 'ecoweb-app' repository. The title bar indicates the target branch is 'feature/ISSUE-35-TinTaeRefactoring' and the destination branch is 'develop'. The dialog has tabs for 'Overview', 'Files' (1), and 'Commits' (2). The 'Title' field contains 'My PR'. The 'Description' field is empty with a placeholder 'Describe the code that is being reviewed'. Below the description is a rich text editor toolbar. The 'Reviewers' field contains 'Orest Deynaka'. There are buttons for 'Add required reviewers' and 'Link work items'.

DevOps, Project management

100% scrum focus

The diagram illustrates the workflow from an Epic to User Stories in Azure DevOps.

Epic: The first screenshot shows the Backlog view for the "ecoweb-ucibox Team". It lists several backlog items, each with a title and a brief description. Two specific items are highlighted with arrows pointing to the "User Stories" section:

- Item 1: Not Functional tasks
- Item 2: Navegation

User Stories: The second screenshot shows the "Boards" view for the "ecoweb-app" project. It displays a Kanban board with columns: To Do, Blocked, 2/s, and Doing. A single user story card is visible in the To Do column:

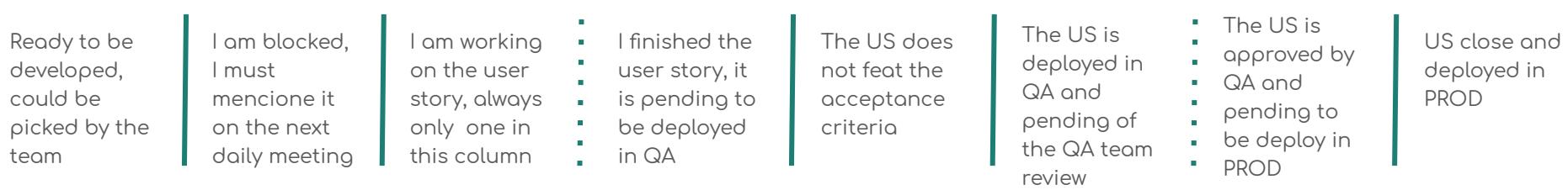
Item ID	Title	State
67	REVISIÓN DE CALCULADORA	To Do

User Stories: The third screenshot shows the "Boards" view for the "ecoweb-app" project. It displays a Kanban board with columns: To Do, Blocked, 2/s, and Doing. A single user story card is visible in the Blocked column:

Item ID	Title	State
35	Ajuste Componente Tarifas (Blocked)	Blocked

DevOps, Project management

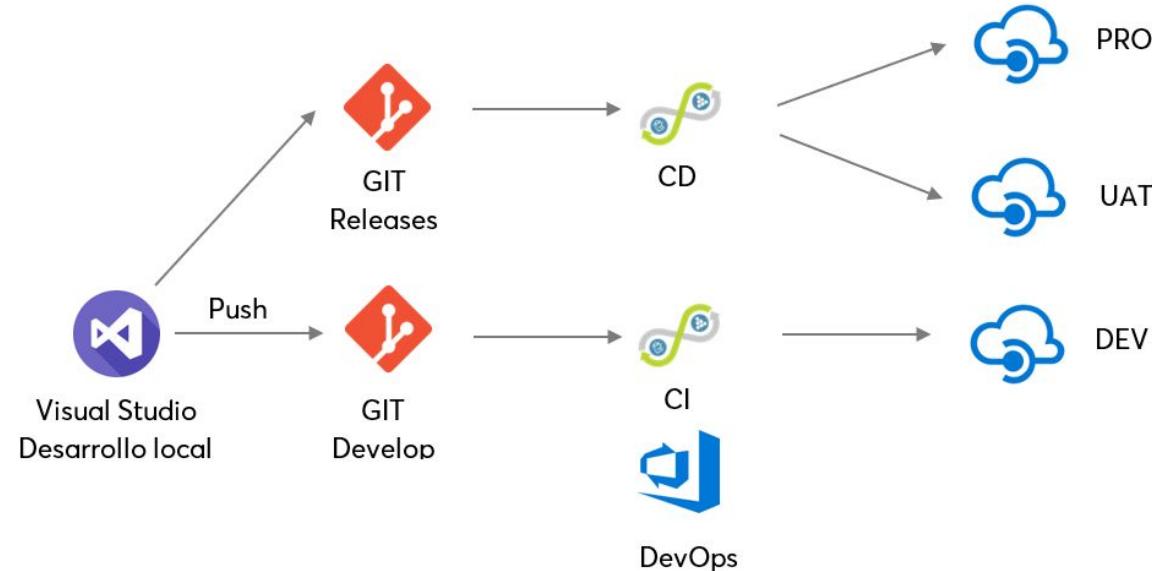
100% scrum focus



To Do	Blocked	Doing	In Error	QA	Pending Prod	Done
<ul style="list-style-type: none">+ New item67 REVISIÓN DE CALCULADORA State ● To Do66 Gestión newsletter State ● To Do65 Formación SITECORE personalización State ● To Do64 Formación SITECORE analítica State ● To Do63 Despliegues automáticos soluciones State ● To Do61 Análisis Azure	<ul style="list-style-type: none">35 Ajuste Componente Tarifas (Blocked) State ● To Do 0/136 Botón dentro de lightbox: State ● To Do	<ul style="list-style-type: none">DoingDone	<ul style="list-style-type: none">0/5	<ul style="list-style-type: none">0/5	<ul style="list-style-type: none">15 Remove the folder Statics mercedes.amoros State ● To Do 1/14 Tin Tae, the update feature in production doesn't work Antoine State ● To Do 0/21 Add URL redirect rules to hipotecas.com mercedes.amoros State ● To Do 1/1	<ul style="list-style-type: none">5 Mortgage application, if the second step is empty, the record does not arrive in CRM Antoine State ● Done 0/1

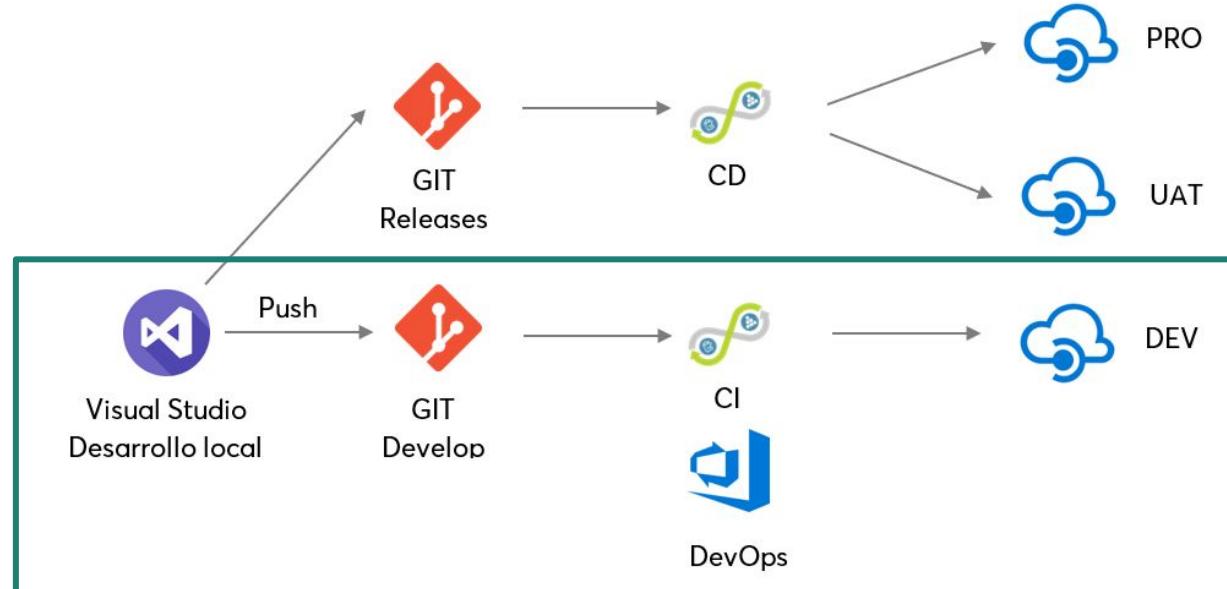
DevOps, CI / CD

Continuous Integration and Delivery Pipeline



DevOps, CI / CD

Continuous Integration and Delivery Pipeline



DevOps, Fractal build pipeline

The image shows three views of the Azure DevOps Pipelines interface:

- Left Panel:** Project navigation for "ecoweb-app". The "Pipelines" item is selected.
- Middle Panel:** Pipeline list titled "Pipelines". It shows "Recently run pipelines":
 - pr-pipeline (Failed)
 - ecoweb-app-pro (Success)
 - ecoweb-app-int (Success)
 - ecoweb-app-fractal (Success)
- Right Panel:** Pipeline details for "ecoweb-app-fractal". The pipeline file "pipeline-fractal.yml" is displayed:

```
1  # Starter pipeline
2  # Start with a minimal pipeline that you can
3  # Add steps that build, run tests, deploy, an
4  # https://aka.ms/yaml
5
6  trigger:
7    branches:
8      - include:
9        - develop
10     paths:
11       - include:
12         - Project/Ecoweb/Ecoweb.Frontend/src
13
14 pool:
15   vmImage: 'ubuntu-latest'
16
17 steps:
18   - task: Npm@1
19     displayName: "Installing npm packages"
20     inputs:
21       command: 'install'
22       workingDir: '$(System.DefaultWorkingDir)'
23
24   - task: Npm@1
25     displayName: "Building fractal"
26     inputs:
27       command: 'custom'
```

A green arrow points from the "ecoweb-app-fractal" pipeline in the middle panel to the corresponding line in the YAML code in the right panel, highlighting the path configuration.

DevOps, Fractal build pipeline

<https://ecoweb-front.azurewebsites.net/>

The screenshot shows the Azure DevOps Pipelines interface for the 'ecoweb-app' project. The left sidebar lists various sections: Overview, Boards, Repos, Pipelines (selected), Pipelines, Environments, Releases, Library, and Task groups. The main area displays the 'Fractal Docs' pipeline, which has three stages: Production (QA), INT (Int), and Stage 1. Each stage is represented by a green circle with initials (A, A, JM) and a release date (20201028..., 20201026..., 20200923...). A large green arrow points from the Stage 1 section towards the right-hand browser window.

The screenshot shows a web browser window displaying a page titled 'Ancho completo: Default'. The page content includes a heading 'Hipotecas para vivir', a subtext 'UCI, un proyecto único donde solo tú eliges lo que quieres: financiación, seguros y servicios para tu casa', and a section 'Condiciones transparentes y sin sorpresas'. Below the content, there is a code editor showing the HTML code for the page, which includes a 'splitter' element. The URL in the browser's address bar is 'ecoweb-front.azurewebsites.net/components/detail/ancho-completo--default.html'.

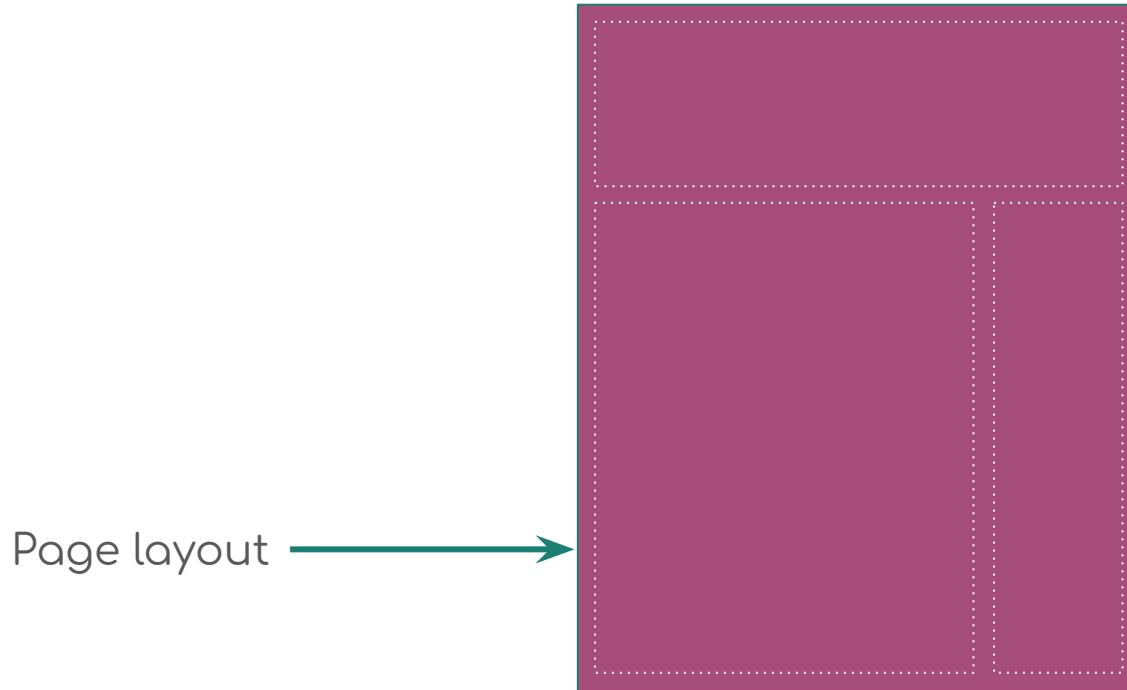
Integration with Sitecore

Sitecore vs Front-end

Different ways to work with Sitecore

- Standard →
 - SXA
 - Headless
- Separation of concept front / back
 - No technological dependencies
 - No structural dependencies
 - Front-end integration step

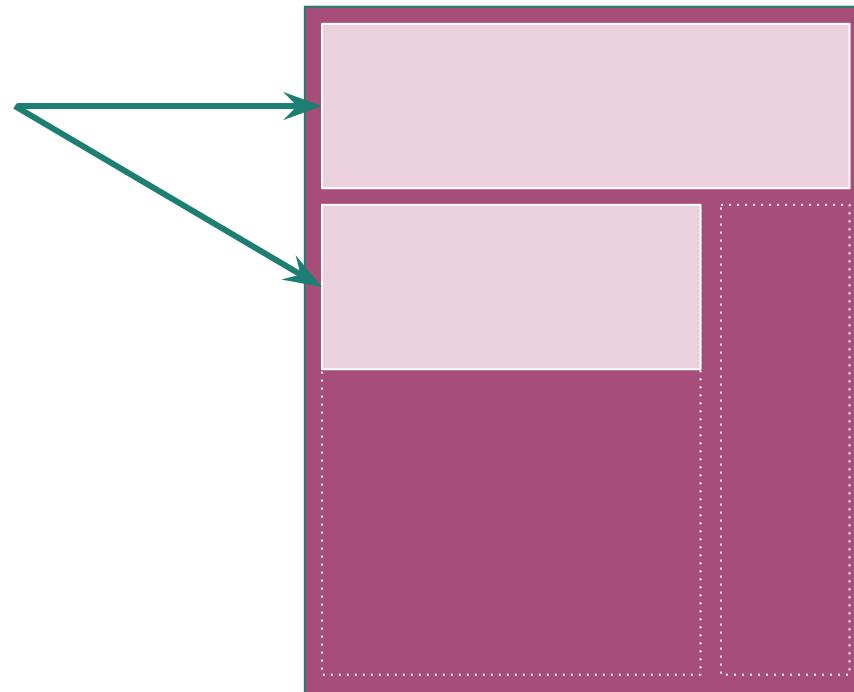
Sitecore rendering and placeholder



Sitecore layout
defines the page
structure with the
outer HTML
mark-up

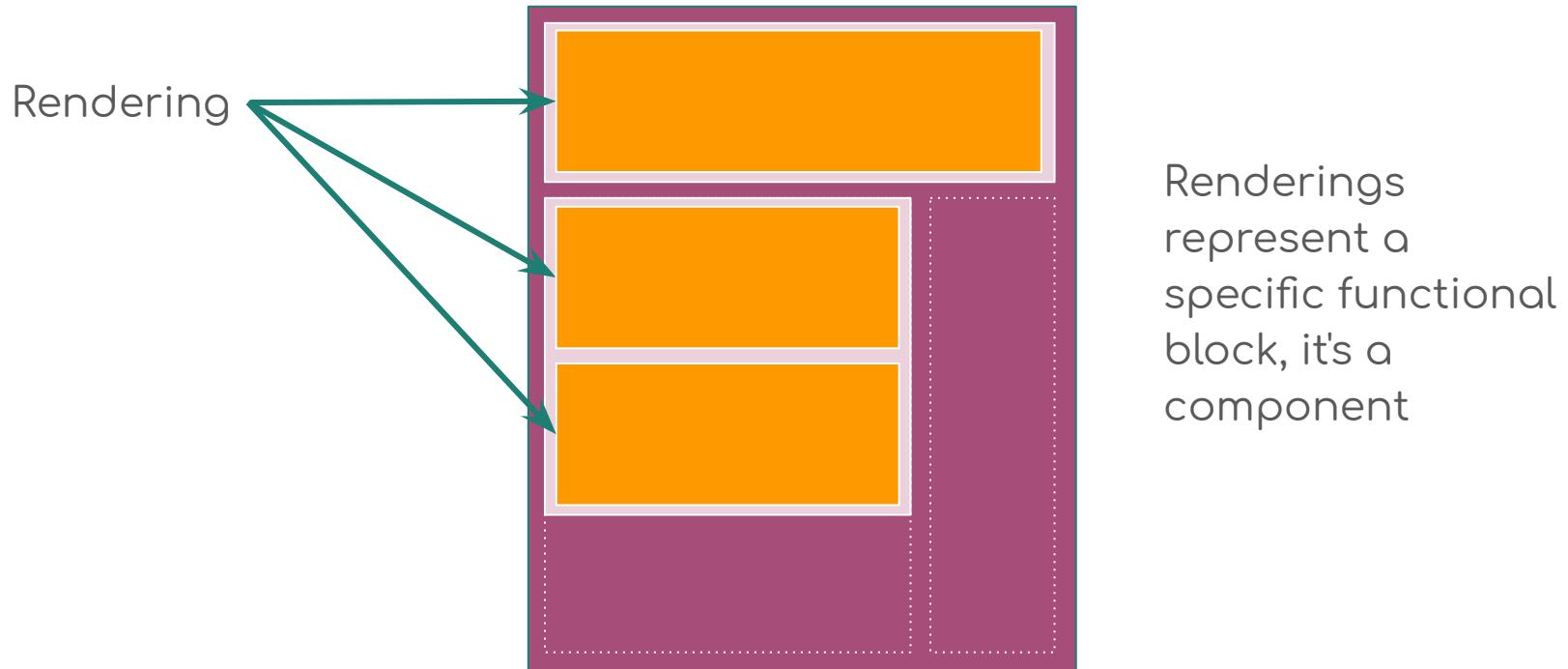
Sitecore rendering and placeholder

Placeholder

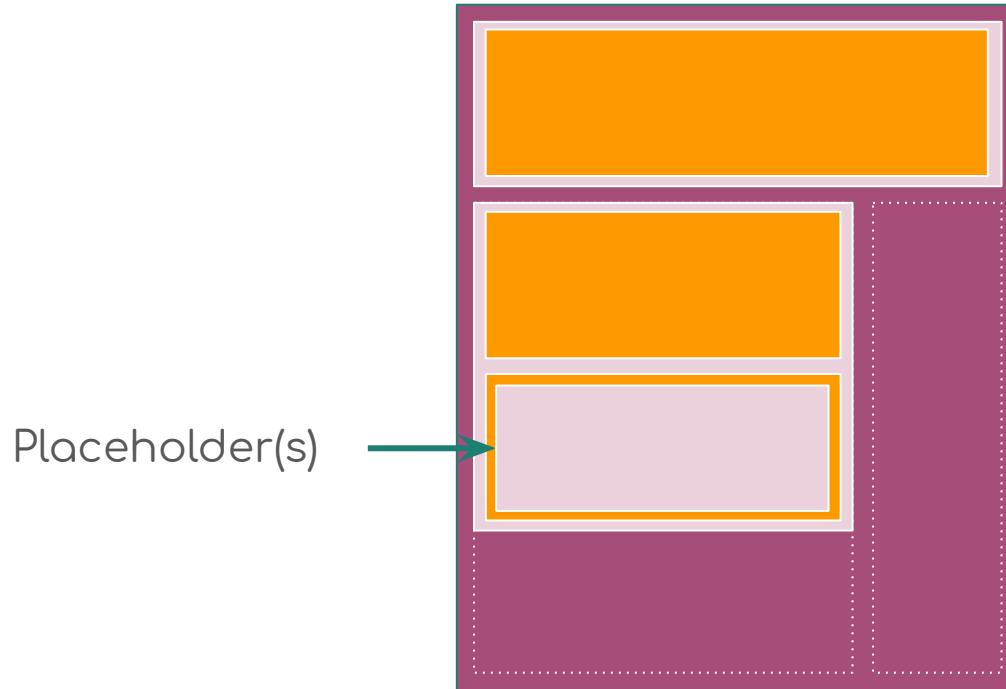


Sitecore
placeholder is a
place where a
rendering can be
bound

Sitecore rendering and placeholder

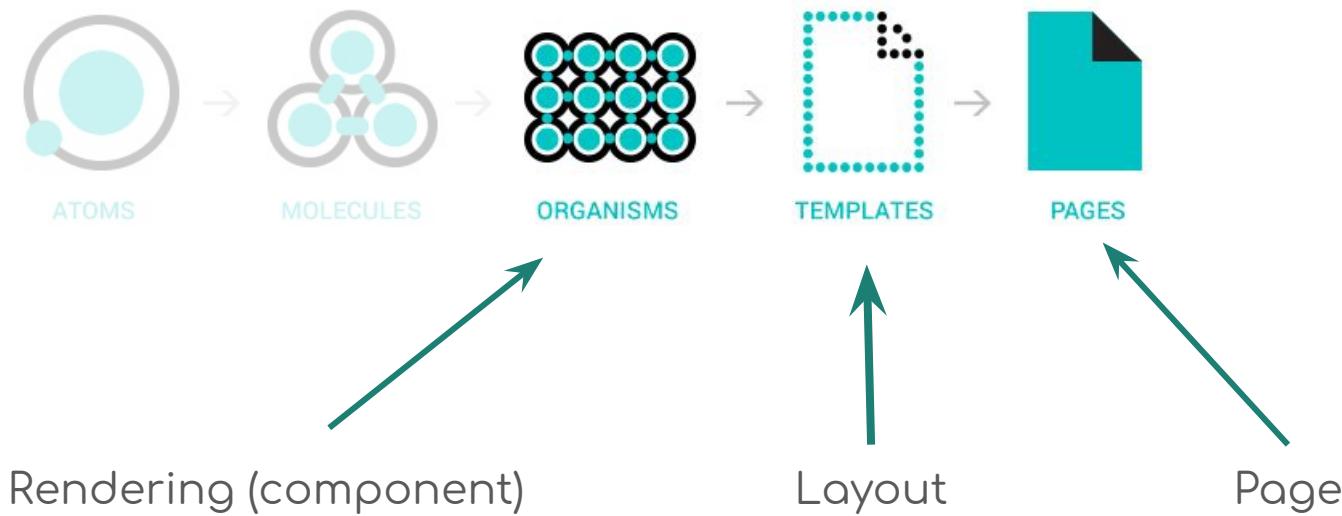


Sitecore rendering and placeholder



A renderings can
have embedded
placeholders

Sitecore vs atomic design

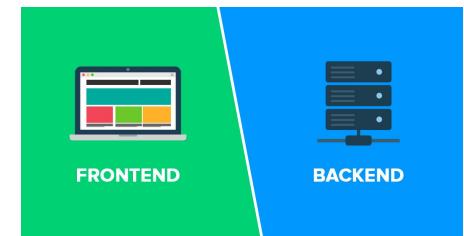


So, what is the integration process?

The integration process consists of integrating the frontend code generated in fractal, into the Sitecore MVC razor files.

For each user story, the process will be:

- ❖ Make an agreement on what kind of elements will be built (layout, component, etc.), their naming (default layout, last post component, etc.), and JS object format if needed.
- ❖ Create frontend sub-tasks to carry out the front-end development: html, css and javascript.
- ❖ Create a back-end sub-task to integrate the frontend code into the MVC razor files. This task can't be executed before the front-end task.
- ❖ The HTML is integrated manually, the CSS, Javascript and other assets are compiled and copied automatically (npm script run during the build pipeline)
- ❖ If the user story has both front-end and back-end subtask, the user story must be assigned to the backend developer
- ❖ The user story will be closed once the front-end is integrated and the user story reviewed by QA in Sitecore. Once closed, any changes or corrections will generate a new user story.



ECOWEB case overview

UCI fractal

- Examples
 - Buttons, Text, Grid, etc.
- Atomic structure
- No templates are used
- Example of pages
- Block section are splitter use cases

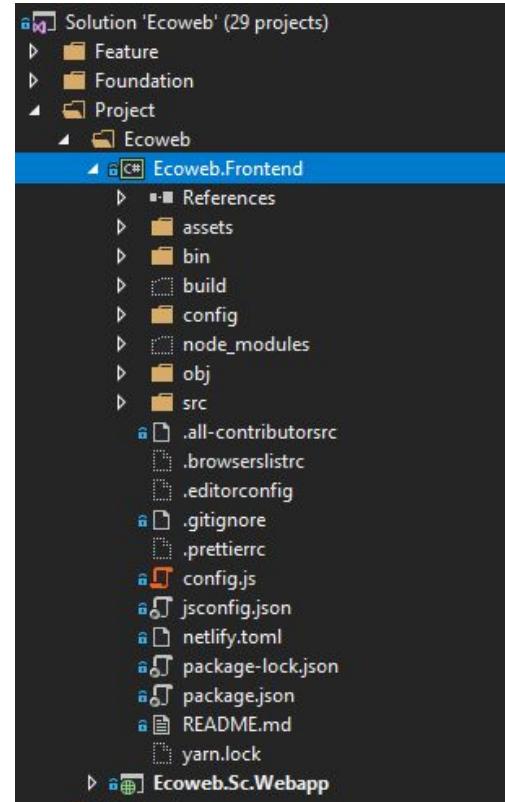
The screenshot shows the UCI fractal interface. At the top, there's a red header bar with a close button (X) and the text "UCI". Below it is a sidebar with a tree-like navigation structure under the heading "COMPONENTS". The "Examples" section is expanded, showing links to "Breadcrumbs Example", "Buttons Example", "Grid Example", "Splitters Example", and "Typography Example" (which is highlighted). Other collapsed sections include "Atoms", "Molecules", "Organisms", "Pages", and "Blocks". To the right of the sidebar, there's a main content area. At the top right of the content area, there's a "View" button. The content area contains a title "Typography example" with a link icon. Below the title, there's a list of typography examples: "Title S (h5)", "Title M (h4)", "Title L (h3)", "Title XL (h2)", and "Title XXL (h1)". Further down, there are "Bodycopy S", "Bodycopy M", "Bodycopy L", "Especial M", and "Text Call to action". At the bottom of the content area, there's a "HTML" section containing the corresponding HTML and CSS code for the "Title S (h5)" example:

```
<style>
  .container>div {
    padding: 5px 0;
  }
</style>

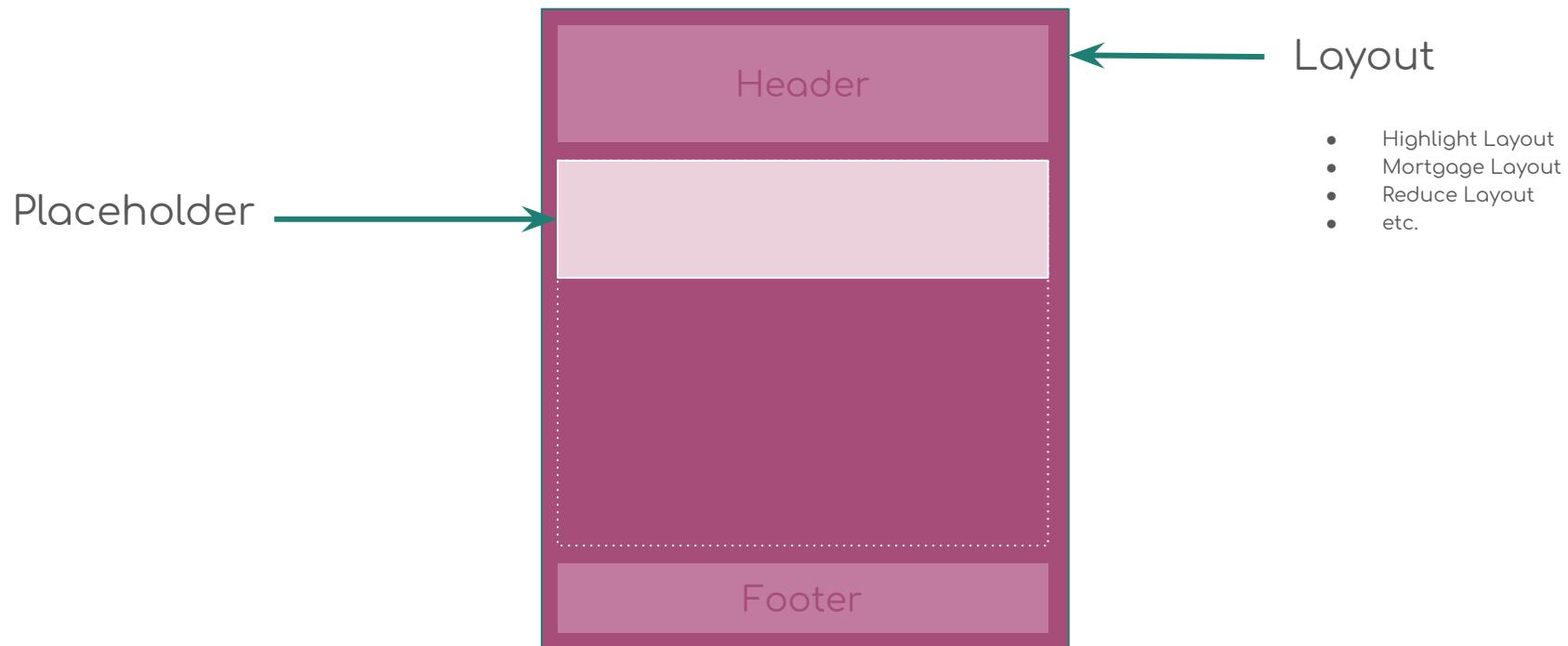
<div class="container">
  <div>
    <h4 class="title-s">
      Title S (h5)
    </h4>
  </div>
</div>
```

File structure

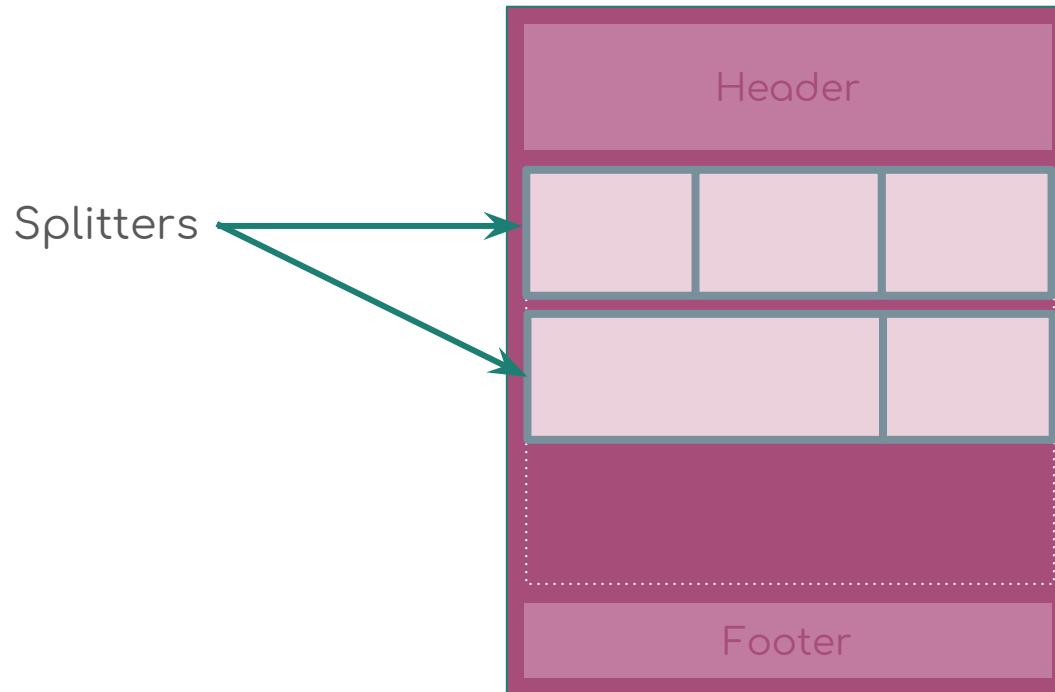
- Same code repository
- Integrated into the VS solution
- npm build script copy the dist folder into the Sitecore ECOWEB project



Sitecore UCI page structure



Sitecore UCI page structure



A splitter is a configurable rendering used to define page body structure on the fly

Sitecore UCI page structure

Splitter are configurable:

- BackgroundColor
- SpacingTop
- SpacingBottom
- ...

The screenshot illustrates the Sitecore User-Configurable Interface (UCI) for managing page structures. On the left, a modal window titled "Control Properties" shows settings for a "Splitter" control. The "Style" tab is selected, displaying options for "BackgroundColor", "SpacingTop", and "SpacingBottom". The "SpacingBottom" dropdown is set to "small". Below the modal, a list of style options includes "Background Color White", "Background Color Accent", "Background Color Accent Dark", "Padding Top Small", "Padding Top Medium", "Padding Top Large", "Padding Bottom Small" (which is highlighted), "Padding Bottom Medium", and "Padding Bottom Large". At the bottom of the modal are "OK" and "Cancel" buttons. To the right, the Sitecore Experience Editor displays a red "One column" content placeholder. A sidebar panel titled "Splitter: Padding Bottom Small" is open, showing the selected style. The "HTML" tab of the sidebar displays the corresponding CSS class: "splitter_padding_bottom_small". The underlying HTML structure is shown as follows:

```
">
<div class="container">
    <div class="splitter__wrapper">
        <div class="row">
            <div class="col-xs-12 col-ms-12 col-sm-12 col-lg-12">
                <div class="colioner placeholder">

```

Not so good

Why don't we have template
(layout) folder ?

The screenshot shows the UCI (User Content Interface) with a red header bar containing a close button and the text "UCI". Below the header is a sidebar titled "COMPONENTS" with the following sections:

- Examples
 - Breadcrumbs Example
 - Buttons Example
 - Grid Example
 - Splitters Example
 - Typography Example (highlighted)
- Atoms
- Molecules
- Organisms
- Pages
- Blocks

The main content area displays a "Typography example" card with the following text and styling:

Title S (h5)
Title M (h4)
Title L (h3)
Title XL (h2)
Title XXL (h1)

Bodycopy S
Bodycopy M
Bodycopy L
Especial M
Text Call to action

HTML

```
<style>
  .container>div {
    padding: 5px 0;
  }
</style>

<div class="container">
  <div>
    <h4 class="title-s">
      Title S (h5)
    </h4>
  </div>
  <div>
    <h4 class="title-m">
      Title M (h4)
    </h4>
  </div>
  <div>
    <h3 class="title-l">
      Title L (h3)
    </h3>
  </div>
  <h2 class="title-xl">
    <b>Title XL (h2)</b>
  </h2>
  <h1 class="title-xxl">
    <b>Title XXL (h1)</b>
  </h1>
  <div>
    <p>Bodycopy S</p>
  </div>
  <div>
    <p>Bodycopy M</p>
  </div>
  <div>
    <p>Bodycopy L</p>
  </div>
  <div>
    <p>Especial M</p>
  </div>
  <div>
    <p>Text Call to action</p>
  </div>
</div>
```

UCI vs Hipotecas

Specific variable and template configuration for each project:

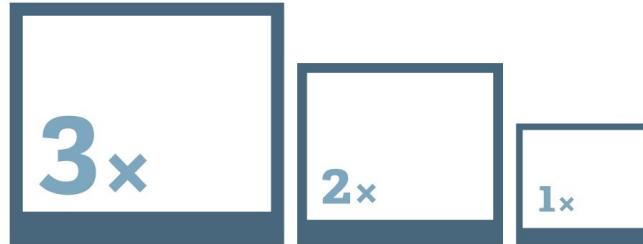
<https://novicell.atlassian.net/wiki/spaces/UCIPRO/pages/1730970042/Frontside+Development>

The image shows two screenshots of the HIPOTECAS.COM website. The top screenshot displays a green-themed landing page with the headline "Damos crédito para que compres tu casa." and a subtext about being specialists who create mortgages according to the user's needs. The bottom screenshot shows a red-themed landing page for "Hipotecas para vivir" with text about UCI being a unique project where users choose what they want, when they want it, and includes financing, insurance, and services. Both pages feature navigation links like "Hipotecas", "Seguros", "Calcula tu hipoteca", "Quiénes somos", and "Te ayudo". The background of the bottom page features a woman smiling and a child in a kitchen.

Real responsive images and lazy load

Addoc development for Sitecore real responsive images:

<https://novicell.atlassian.net/wiki/spaces/UCIPRO/pages/1745158586/Lazy+loading+and+responsive+images>



Sitecore extension:
\Foundation\Ecoweb.Sc.Foundation.Helpers\code

```
namespace Ecoweb.Sc.Foundation.Helpers
{
    public static class HtmlMediaHelpers
    {
#pragma warning disable CA1717 // Only FlagsAttribute enums should have plural names
        public enum Resolutions...
        ...
        public static IHtmlString BackgroundImage(this HtmlHelper _, string url, Dictionary<string, object> attributes)
        {
            ...
        }
    }
}
```

```
<div
    class="access-highlight lazyload"
    data-bg="/dist/images/access-highlight-bg-mob.jpg 767px,
             /dist/images/access-highlight-bg-tab.jpg 1023px,
             /dist/images/access-highlight-bg-lap.jpg 1279px,
             /dist/images/access-highlight-bg.jpg">
```

QA

Merci

antu@ngiro.net