Spike to evaluate the viability of replacing the Font Awesome icons library with Phosphor

Goal

The present spike pretends to evaluate the possibility of replacing the actual icons library (Font Awesome) with a new one (Phosphor).
Note: there are some Prime Faces Icons being used too (.

Font Awesome (main-frontend):

We currently have 473 results of FA icons in HTML templates (no PrimeFaces). These can be separated by styles (4 styles being used):
fa (classic), fas (solid), far (regular), fal (light)

Font Awesome (admin-frontend):

We currently have 56 results of FA and 42 PrimeFaces icons in html templates. Same styles as in main-frontend

General considerations:

Some icons are being declared in .scss files (inside "::before" pseudo-elements) that will also need replacing
Font awesome variable declaration files are being used which might make migration a bit complex (see src/assets/scss/fontawesome/_variables.scss as an example), as those variables refer to Unicode values that translate into icons. This variables are then used in the .SCSS files mentioned above as content values. Some elements, such as the search input present in the header component (main-frontend), implement their icons in this way.
Phosphor doesn't seem to support a direct replacement for this approach (the unicode value can be manually searched in the package CSS files, but is far from ideal. This might change in the future, as the library devs mention in this link).

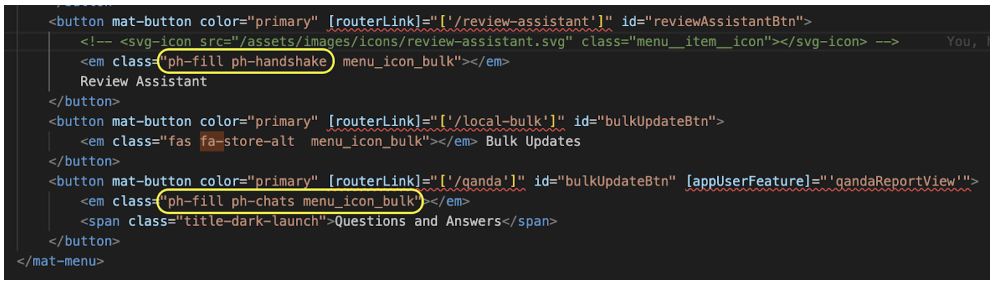
Phosphor Considerations:

- Phosphor is open source, all the content is available
- Phosphor does not provide native support for Angular, the web version has to be used
- Comes in 6 weights: regular, thin, light, bold, fill, and duotone. The package can be installed via npm: **npm i @phosphor-icons/web**
- Imports have to be done individually by weight in the main SCSS file (if someone knows a better way, just leave a comment):

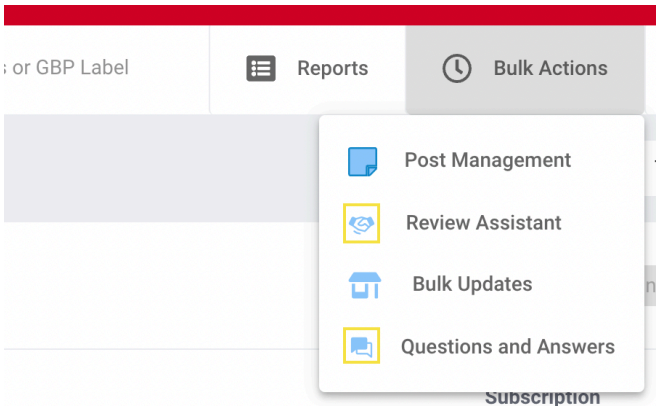
- All weights use the :before pseudoelement to inject the font glyph (same as fontawesome), so overriding this property in icon classes can break them.
- Directly replacing icons shows a size difference with font awesome. This will require a slight adjustment

Initial tests

A small set of icons were successfully replaced in the header.component, by directly replacing the font awesome classes inside the HTML elements, with phosphor classes:



Changes in the front end (new icons being rendered):



Direct replacements in the HTML code, such as the ones above didn't prove to be hard, they should be pretty straightforward. It's worth mentioning that SVG images can be directly replaced with Phosphor icons, such as in the first example of the first screenshot.

It's a different story when dealing with icons defined inside the pseudo-element ::before, such as the magnifying glass in the search input. In this case, because the declaration of the Unicode

value and its properties (margins, size, color) are being done in the CSS file, a direct implementation of the phosphor icon won't suffice.

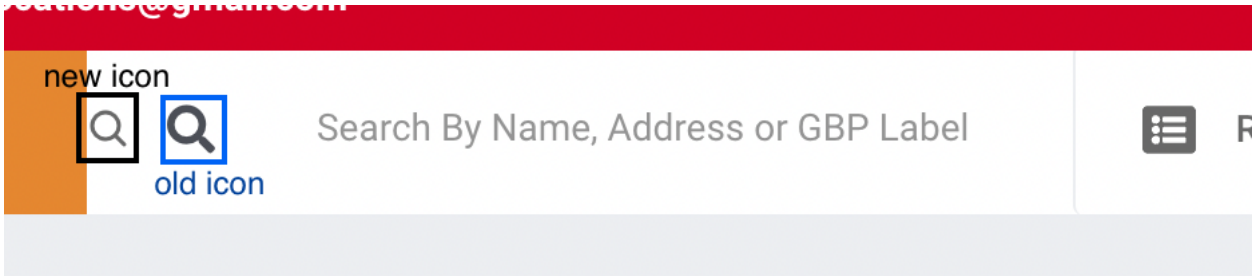
Original implementation:

```
61
62    &--header {
63      position: relative;
64      &::before {
65        content: '\f002';      Samuel Morhaim, hace 2 años
66        font-family: "Font Awesome 5 Pro", sans-serif;
67        font-size: 18px;
68        font-weight: 700;
69        position: absolute;
70        left: 30px;
71        top:  50%;
72        transform: translateY(-50%);
73        @extend .txt--gray;
74      }
75      input {
76        height: $header-height;
77        padding-left: 70px;
78      }
79    }
```

Addition of new phosphor icon in the HTML template:

```
<div class="input-field input-field--search input-field--header" *ngIf="auth?.session?.role != 'member'">
  <i class="ph-bold ph-magnifying-glass"></i>
  <input type="text" id="searchHeader" matInput placeholder="Search By Name, Address or GBP Label" (input)="handleChangeSearch($event)"
  [matAutocomplete]="auto"
  [formControl]="selectedSearch">
  <mat-autocomplete #auto="matAutocomplete" [displayWith]="display" (optionSelected)="selected()"
                    class="mat-autocomplete-panel--dimensions-global-search">
```

Result:



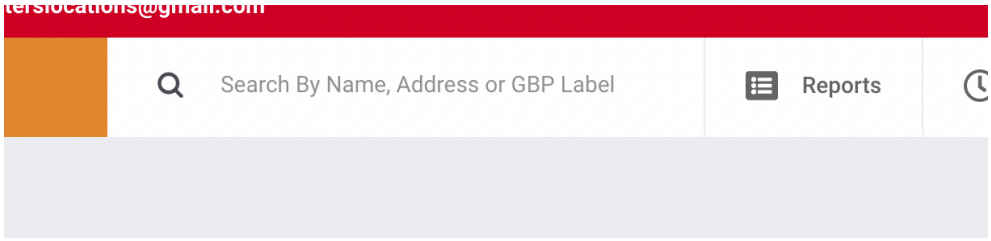
As shown in the screenshot above, further adjustments will be needed to correctly display the new icon (CSS properties).

A different approach would involve replacing the Unicode value of the older icon, with the new one, which requires (as of today) searching the corresponding Phosphor style CSS file to find the new icon's Unicode.

```
2226  .ph-bold.ph-magnifying-glass:before {
2227    content: "\ebdd";
2228  }
```

Directly replacing the old value (‘\f002’) with the new one in the pseudo-element declaration, and changing the font family to the correct one, results in the following:

```
62  &--header {
63    position: relative;
64    &::before {
65      content: '\ebdd';
66      font-family: "Phosphor-bold", sans-serif;
67      font-size: 18px;
68      font-weight: 700;
69      position: absolute;
70      left: 30px;
```



This approach, although inefficient, proves to deliver the desired results.

However, it’s worth mentioning again that Unicode values are unstable (as mentioned in this thread), at least until the new Phosphor version comes out with stabilized icon codepoints. This means that there is a chance that if we eventually update the library to its latest version, some of those codes might change, breaking the styles and requiring fixing.

Conclusion

The replacement is possible but requires some unstable approaches. It’s a matter of deciding if the benefits outweigh the risks (minimal, but existent) and required implementation times.