CODING STANDARD

1. Use of **comments** (use snippets) whenever it feels necessary within the code. In classes, functions, documents, internal code, etc ...

Ex:

/\*\*

\* Auxiliar function to manage and retrieve product description

\* from the products page and the size of the linked HTML

\* @param $link

\* @return array

\*/

private function getProductExtraInfo($link) {

// Curl url and retrieve page HTML.

$req = $this->getData($link);

$crawler = new Crawler($req['content']);

$data = [];

// Apply css filter to extract specific css class element.

$data['description'] = trim($crawler->filter('.productText')->first()->text();

$data['size'] = $req['request\_size'];

return $data;

}

1. **Readability, format and spacing.** Make the code easy to read, use tabs for spacing.

Ex:

if ($hours < 24 && $minutes < 60 && $seconds < 60) {

return true;

}

else {

return false;

}

1. The naming of the file names will be done with the format **snake\_case**, the **UpperCamelCase** classes and those of the rest of the variables, functions, etc. will use the **lowerCamelCase.**
2. All denominations, documentation and comments will be in the **English language**.