Colección de herramientas y recursos del equipo de polymer para crear apps. Incluye:

1. Elementos layout
2. Routing
3. Localization (soporte para idiomas)
4. Almacenamiento
5. Polymer CLI (visto en una lección anterior)

Elementos layout

Estos elementos nos permiten usar 'piezas' típicas que toda buena app debería tener de forma sencilla, incluyen:

- app-drawer : barra lateral
- app-drawer-layout : app con barra lateral
- app-header : cabeceras configurables
- app-header-layout
- app-scrollpos-control: control del scroll
- app-toolber: barras de herramientas
- app-box: contenedor de la app ideal para arquitecuras app-shell

Routing

Gracias a los elementos `app-location` y `app-route` . Veamos un ejemplo:

```
<app-location route="{{route}}"></app-location>
  <app-route
      route="{{route}}"
      pattern=":page"
      data="{{routeData}}"
      tail="{{subroute}}"></app-route>

  <app-drawer-layout fullbleed>

    <!-- Drawer content -->
    <app-drawer>
      <app-toolbar>Menu</app-toolbar>
      <iron-selector selected="[[page]]" attr-for-selected="name" class="drawer-list" role="nav
igation">
        <a name="view1" href="/view1">View One</a>
        <a name="view2" href="/view2">View Two</a>
        <a name="view3" href="/view3">View Three</a>
      </iron-selector>
    </app-drawer>
```

```
  </app-drawer>

  <!-- Main content -->
  <app-header-layout has-scrolling-region>

    <app-header condenses reveals effects="waterfall">
      <app-toolbar>
        <paper-icon-button icon="menu" drawer-toggle></paper-icon-button>
        <div title>My App</div>
      </app-toolbar>
    </app-header>

    <iron-pages role="main" selected="[[page]]" attr-for-selected="name">
      <my-view1 name="view1"></my-view1>
      <my-view2 name="view2"></my-view2>
      <my-view3 name="view3"></my-view3>
    </iron-pages>

  </app-header-layout>

</app-drawer-layout>
```

Y su js:

```
Polymer({

    is: 'my-app',

    properties: {

      page: {
        type: String,
        reflectToAttribute: true,
        observer: '_pageChanged'
      },

    },

    observers: [
      '_routePageChanged(routeData.page)'
    ],

    _routePageChanged: function(page) {
      this.page = page || 'view1';
    },

    _pageChanged: function(page) {
      // load page import on demand.
      this.importHref(
        this.resolveUrl('my-' + page + '.html'), null, null, true);
    }

  });
```

Localization

Gracias al app-localize-behavior veamos un ejemplo de uso:

Necesitamos el behavior:

```
behaviors: [
    Polymer.AppLocalizeBehavior
],
```

Cargar nuestro archivo de traducciones:

```
attached: function() {
  this.loadResources(this.resolveUrl('langs/my-app.json'));
},
```

Un ejemplo de traducciones puede ser:

```
  {
      "es":{
          "menu": "Menu",
          "what-is": "Qué es GDGKids?",
          "5-commandments": "Nuestros 5 mandamientos",
          "events": "Eventos",
          "faq": "FAQ"
      },

      "en":{
          "menu": "Menu",

          "what-is": "What is GDGKids?",
          "5-commandments": "Our 5 commandments",
          "events": "Events",
          "faq": "FAQ"
      }

  }
```

Para traducir finalmente hacemos por ej:

```
<a name="what-is-it" href="/what-is-it">{{localize('what-is')}}</a>
```

** Fijaos el buen uso que se hace aquí de un computed binding :)

Storage

Para aprender más de indexeddb y el almacenamiento local en web apps visitad este enlace: https://elements.polymer-project.org/elements/app-storage?active=app-indexeddb-mirror