Curso Polymer Softtek 

Julio 2019

1. **Introducción**

**Objetivo del curso:**

* Aprender los fundamentos de CSS, Javascript, HTML5 que son tema prerrequisito para desarrollo con Polymer
* Aprender Polymer para crear componentes de la capa Front End web. Estos componentes son responsivos, modulares y encapsulados lo que permite crear aplicaciones reutilizables y más fáciles de mantener.

**Dirigido a:**

* Programadores de aplicaciones Web que necesiten aprender a desarrollar con Polymer y Cells

**Prerequisitos**

* Conocimientos sólidos en programación
* Conocimientos básicos de desarrollo web con javascript, css, html.
* Conocimientos básicos en conceptos de orientación a objetos y principios de diseño.

**Notas:**

* Todo el material de consulta esta referenciado a links el internet. Lo que se proporciona aquí es la guía de temas a cubrir para aprender.
* Los alumnos que no puedan estar en el curso presencial deberán usar el temario como guía de autoestudio.

1. **Temario**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Temas | Horas | Planeación |
| 0 | **Introducción y Ambientación**   * Objetivo del curso * Prerequisitos   + Javascript, HTML, Css básico * Reglas del curso:   + Forma de trabajo colaborativo: entrega individual, compartir conocimiento.   + Investigación por su cuenta (No todo va a estar digerido)   + Presentar su mejor trabajo. En cada trabajo manejar casos de error.   **Ambientación**  Instalar las siguientes herramientas   * Atom, plugins: file-icons, emmet, color-picker, highlight-selected * Notepad * Node JS * Git * Package manager: npm, bower * Chrome Dev Tools   + Plugins postman,   + Web server for Chrome   + Debugger, paneles de application, cache, shadow dom, service worker   Lectura:  <https://www.careerride.com/online-practice-test/Oops>  <https://escuela.it/cursos/curso-de-polymer-2/clase/web-components-v1-y-polymer-2> | Día 01  4 horas |  |
| 1 | **Orientación a Objetos**   * Principios de Diseño   + High Cohesion, Low Coupling, Open Closed, Inversion of Control, Interface Segregation, Single Responsibility, Composition over Inheritance, Liskov * Conceptos de Orientación a objetos   + Encapsulación, Polimorfismo, Herencia, Abstracción * Desarrollo siguiendo principios SOLID   **Lecturas / Ejemplos:**  **Ejercicios:**  **Laboratorio:**  <https://www.goconqr.com/p/5819053-quiz---object-oriented-javascript-quizzes>  <https://www.careerride.com/online-practice-test/Oops> |  |  |
| 2 | **HTML5, CSS**   * Conceptos   + Progressive Web Applications, Principios Diseño SOLID * Desarrollo de aplicaciones web   + Ciclo de Request Response   + Desarrollo de 3 capas * Estructura documento HTML5, Que es CSS? Que es Javascript? * Ejercicios   + Folder 01Html5   + HTML5, Javascript   + Página HTML5 con separación de js, css * Laboratorio: Creación de una página básica   + Crear una página con navegación   **Lecturas / Ejemplos:**  **Ejercicios:**  **Laboratorio:**  Referencias:  <http://www.cs.sjsu.edu/faculty/pearce/modules/lectures/web/javascript/dom.htm>  <https://www.w3schools.com/html/html5_intro.asp>  <https://www.w3schools.com/css/>  <http://caniuse.com>  <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>  <https://dillinger.io/> | Día 01  4 horas |  |
| 3 | **Javascript Básico**   * Javascript basico: Sintaxis, Tipos de variable, Operadores, Scope, Ciclos, Objetos, API, Clases, DOM, funciones, arreglos, eventos, Excepciones, Parseo * Debugger, consola. * Ejercicios de lógica de programación * Ejercicios   + Ejemplo de página para modificar elementos de HTML   + Tipo de cambio Pesos USD   + Crear objeto persona con datos capturados desde pantalla   + Cifrado Cesar   + Validación de correo   + Elementos repetidos   + Saber si un arreglo tiene elementos repetidos   + Saber si 2 números en un arreglo pueden sumar un numero   + <https://www.testdome.com/questions/javascript/date/8521?visibility=1&skillId=2> * Laboratorio   + Testdome   + HackerRank   **Lecturas / Ejemplos:**  **Ejercicios:**  **Laboratorio:** | Día 02  8 horas |  |
| 4 | **Javascript Avanzado**  **ECMAScript6**   * Uso de $, $$, \_ en métodos y propiedades. * Expresiones regulares * Clases * JSON * Funciones incorporadas y de utilería. * Inline functions, arrow functions, eventos, prototipos, encadenamientos, callbacks * Clojures**,** Service Workers * Fetch, Promesas, * IntervalosyTimers * Ajaxy REST * Extra: Cookies, Local Storage, Session Storage * Ejercicio   + Crear Todo App + Remove completed Button * Laboratorio   + Agenda que agrega y consulta contactos   + Extra: Cookies, Local Storage, Session Storage * Examen:   + Ejercicios Testdome de Javascript   **Lecturas / Ejemplos:**  **Ejercicios:**  **Laboratorio:**  Referencias:  <http://es6-features.org/#StatementBodies>  <https://developers.google.com/web/updates/2015/03/introduction-to-fetch>  Regexp  <https://www.w3schools.com/jsref/jsref_obj_regexp.asp>  <https://www.regexpal.com/> | Día 03  Dia 04  16 horas |  |
| 5 | **Responsive Design**   * CSS3, Selectores, layouts, Media queries * Flexbox * Grid layout * Agregar Css a HTML * Cambiar CSS con Javascript * Ejercicio   + CSS   + Tutoriales Css Garden   + Ranita   + Grid * Laboratorio   Links:  <http://cssgridgarden.com/>  <https://flukeout.github.io/>  <https://flexboxfroggy.com/#es>  <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>  Responsive patterns  <https://developers.google.com/web/fundamentals/design-and-ux/responsive/patterns>  **Lecturas / Ejemplos:**  **Ejercicios:**  **Laboratorio:** | Día 05  5 horas |  |
| 6 | **HTML5 Custom Elements**   * Shadow Dom * Clases * Metodos callback * Custom Events * Ejercicios   + x-counter   + x-cesarcipher * Laboratorio   + Agenda   **Links:**  <https://www.html5rocks.com/en/tutorials/webcomponents/customelements/>  **Lecturas / Ejemplos:**  **Ejercicios:**  **Laboratorio:** | Día 06  8 horas |  |
| 7 | **Polymer 2.0 Basico**   * Ambientación   + Polymer client   + Creación elemento, app, starter kit, shop * Conceptos   + Properties   + Ciclo de vida   + Data binding   + Observers   + Computed values   + Eventos   + Shadow DOM * Elementos   + Iron elements, paper elements, API elements * Utilerias   + iron-ajax, iron-form,   + dom-repeat, dom-if, dom-bind, filtering and sorting   + Lectura de json desde archivo * Ejercicios * Laboratorio   **Lecturas / Ejemplos:**  **Ejercicios:**  **Laboratorio:**  **Links:**  **Introducción**  <https://bbvaopen4u.com/es/actualidad/web-components-presente-y-futuro-en-el-desarrollo-web>  <https://platzi.com/blog/web-components-polymer/>  <https://developers.google.com/web/fundamentals/architecture/app-shell>  **Referencia**  <https://codelabs.developers.google.com/>  https://polymer-library.polymer-project.org/2.0/docs/devguide/  **Laboratorios**  Tu primer app progresiva  <https://codelabs.developers.google.com/codelabs/your-first-pwapp/index.html?index=..%2F..index#0>  Polymer First Element  <https://codelabs.developers.google.com/codelabs/polymer-first-elements/index.html?index=..%2F..index#2>  Adivina Bandera  <https://codelabs.developers.google.com/codelabs/whose-flag/index.html?index=..%2F..index#0>  Migración de App  <https://codelabs.developers.google.com/codelabs/migrate-to-progressive-web-apps/index.html?index=..%2F..index#0>  Starter Kit app  <https://codelabs.developers.google.com/codelabs/polymer-es2015/index.html?index=..%2F..index#0>  <https://meowni.ca/posts/polymer-2-cheatsheet/> | Día 07  8 horas |  |
| 8 | **Polymer 2.0 Avanzado**   * Shadow DOM & styling * Shadow DOM concepts * DOM templating * Custom CSS properties * Manejo de configuracion desde archivos * Complex observers   + Actualizar estilos desde Polymer this.updateStyle()   + this.set(), this.splice(), this.push() * Aplicaciones   + App templates   + Routing   + Localization   + App storage   + Service worker   **Lecturas / Ejemplos:**  **Ejercicios:**  **Laboratorio:** | Dia 8  8 horas |  |
|  |  |  |  |
| 10 | **Proyecto o Examen Final**   * Creación de una aplicación completa Agenda   + Agenda con estilos, imágenes, paper elements, con guardado en BD fractal   + Si se vio el tema de Scrum, aplicarlo   Examen:  <https://glitch.com/edit/#!/delicious-beard?path=frog-component.html:1:4> | Día 10  Día 11  8 horas |  |
| 11 | **Precertificación**   * Simulación de examen teórico en Hacker Rank | Día 12  2 horas |  |
| A | **Integración con backend, Web Services, Bases de datos**   * Metodos HTTP * Integracion con una app java mediante WS REST * Integración con Firebase * Ejercicio * Laboratorio   **Lecturas / Ejemplos:**  **Ejercicios:**  **Laboratorio:**  <https://www.tutorialspoint.com/firebase/>  <https://desarrolloweb.com/articulos/introduccion-firebase-backend-nube.html> | Día 9  8 horas |  |
| B | **Tema Opcional**  **Integración con backend, Web Services, Bases de datos**   * Metodos HTTP * Integracion con una app java mediante WS REST * Integración con Firebase * Ejercicio * Laboratorio   <https://www.tutorialspoint.com/firebase/>  <https://desarrolloweb.com/articulos/introduccion-firebase-backend-nube.html> | Día 9  8 horas |  |
| C | **GIT**  **Tema Opcional**  Este Tema se deja como opcional porque no hay permiso de usar un repositorio Git   * Funcionamiento de Git * Comandos * Creación de cuenta en GitHub * Trabajo con Repositorios remoto y local * Laboratorio   Links:  <https://github.com/>  <https://try.github.io/> | 4 horas |  |
| D | **Scrum**  **Tema opcional**  Desarrollo de un proyecto Polymer bajo la metodología Scrum  **Ejercicio Scrum**  **Proyecto Codigos postales**  3 componentes:   * Campos de búsqueda y tabla de resultados * Componente de consulta de códigos postales json * Backend de información * Clase y estructura de datos   Incluir una actividad de peer review de prototipo  Al final intercambiar el componente de consulta para validar que es modular  **Codigos postales**  <https://github.com/Munett/API-Codigos-Postales>  <https://api-codigos-postales.herokuapp.com/v2/codigo_postal/09880>  <https://github.com/redrbrt/sepomex-zip-codes> | 8 horas |  |
| E | **Temas extras opcionales:**  **Extras**   * Pollyfills * Gesture events * Apache Cordoba * Cambios en Polymer 3 * Seguridad * Uso de Sesiones | ¿? horas |  |

**Equipo requerido para impartir la clase**

* Laptop por cada participante con al menos 4GB de RAM con permisos de administrador, WiFi habilitado.
* Software :
  + Node JS versión 8.11 (que incluye npm)<https://nodejs.org>
  + Git versión 2.17<https://git-scm.com/downloads>
  + Atom versión 1.27<https://atom.io/>
* Monitor ó proyector, pizarron,
* Acceso a todos los sitios listados es INDISPENSABLE

**Metodología**

* Presentaciones
* Ejemplos
* Ejercicios
* Laboratorio
* Investigación por su cuenta

**Software:**

7zip

Notepad++

Git

NodeJs

Atom

**Acceso a recursos y subrecursos en internet:**

**Sugerencias para el instructor**

Las instalaciones deben tener acceso a todos los links listados al final del documento

**NOTAS**

**Día 01**

3:02 PM

<https://brianur.info/cifrado-caesar/>

var ascii = "cadena".charCodeAt(0);

var car = String.fromCharCode(65);

var $ = document.getElementById;

var myElement = $('targetElement');

**Laboratorios**

Tu primer app progresiva

<https://codelabs.developers.google.com/codelabs/your-first-pwapp/index.html?index=..%2F..index#0>

Polymer First Element

<https://codelabs.developers.google.com/codelabs/polymer-first-elements/index.html?index=..%2F..index#2>

Adivina Bandera

<https://codelabs.developers.google.com/codelabs/whose-flag/index.html?index=..%2F..index#0>

Migración de App

<https://codelabs.developers.google.com/codelabs/migrate-to-progressive-web-apps/index.html?index=..%2F..index#0>

Starter Kit app

<https://codelabs.developers.google.com/codelabs/polymer-es2015/index.html?index=..%2F..index#0>

**Ambientación**

# Instalacion de SW base y Verificacion de productos instalados

* Node

node –v

* Atom
* Web Server Chrome
* Git

npm –v

git --version

# Instalacion de herramientas de Polymer para ambiente de Polymer

npm install -g bower

npm install -g polymer-cli

git config --global –edit

# Configurar esta variable para

set PATH=C:\Users\jose.iturbide\AppData\Roaming\npm;%PATH%

set ANDROID\_SDK=C:\Users\jose.iturbide\AppData\Local\Android\Sdk

../bin/sdk licenses

bower install jquery

bower install --save Polymer/polymer

polymer init

# Para Polymer 3.0

polymer serve --npm

polymer serve --open

open http://127.0.0.1:<port>/demo/polymer serve --npm

open http://127.0.0.1:<port>/demo/

**Para ejecutar un ejemplo descargado desde Git.**

git clone https://github.com/PolymerLabs/polymer-2-first-element.git

cd polymer-2-first-element

bower install

# Instalar

npm install –g python-shell

npm install –g python

**# Install Gulp**

npm install –g gulp-cli

npm install gulp –D

npx -p touch nodetouch gulpfile.js

gulp –help

Polymer 3

**NO** npm install --save PolymerElements/paper-button

npm install --save @polymer/paper-button

npm install --save @polymer/iron-demo-helpers

#Cordova

npm install -g cordova

Firebase

* Cuidar API key
* Usar realtime o base
* <https://picsum.photos/300/300?random>
* Regexp Test

npm install --save-dev sw-precache

<https://bbva-files.s3.amazonaws.com/cells/bbva-catalog/index.html#/?v=1.7.0>

Firebase

bower install firebase

npm install firebase –save

Update available 5.5.1 → 6.9.0

npm i -g npm