**INTRODUCTION TO WEB SERVICES**

* **Define the basic requirements of your project**

Web platform, with oriented to documents database that will hold information about votant users, independent candidates and social initiatives. Will give people the facility

to get information, filtered, about candidates, initiatives and administrative campaigns and signing on to a social initiative via internet.

* **Design the database of your project.**

**Usuario:**

* ID
* Nombre(s)
* Apellido paterno
* Apellido materno
* Email
* Contraseña / (Opción de vincular con fb)
* Dirección
* Calle y número
* Número interior
* Código Postal
* Colonia
* Id de municipio o población / pertenece a un municipio
* Id de estado / pertenece a un estado
* Número de tel celular
* Número de tel casa
* Id de INE / Tiene una INE
* Sube su credencial de elector y firma digitalizada.
* Puede apoyar candidatos independientes e Iniciativas de ley
* Compartir foto consentimiento
* Subir foto consentimiento

**Identificación:**

* OCR
* CE
* NE
* Id de municipio o población / pertenece a un municipio
* Id de estado / pertenece a un estado
* Id de sección
* Id del usuario / Pertenece a un usuario
* Foto frontal
* Foto posterior
* Foto de firma
* Foto de consentimiento

**Candidatos:**

* Federales:

1. 3 secciones: Senador / Diputado federal / Presidente de la República
2. Diputados federales pueden ser filtrados por Estado y Distrito uninominal federal.
3. Senadores pueden ser filtrados por estado.

* Locales:

1. 3 secciones: Gobernador / Presidente municipal / Diputado local
2. Todas las secciones se pueden filtrar por estado.
3. Presidente municipal se puede filtrar por municipio.
4. Diputado local puede filtrarse por municipio y distrito local.

* ID
* Tipo de candidato: Federal / Local
* Id de la sección / pertenecen a una sección
* Declaración patrimonial
* Declaración de intereses
* Declaración fiscal
* Tiene muchos apoyos de iniciativas

**Sección:**

* ID
* Nombre de la sección : Senador / Diputado federal / Presidente de la República Gobernador / Presidente municipal / Diputado local
* Id del estado, si pertenece a alguno
* Id del municipio, si pertenece a alguno
* Id del distrito, si pertenece a alguno
* Tiene muchos candidatos

**Distritos:**

* ID
* Tipo: Uninominal Federal / Local
* Id del estado / Pertenece a un estado
* Número de distrito relativo al estado
* Id del distrito federal, si es que pertenece a alguno (es decir, si es distrito local)
* Tiene muchos distritos locales, si es distrito federal
* Tiene muchos municipios
* Tiene muchas secciones

**Estados:**

* ID
* Nombre
* Tiene muchos distritos federales
* Tiene muchos distritos locales
* Tiene muchos municipios
* Tiene muchas secciones

**Municipios:**

* ID
* Nombre
* Id del estado / Pertenecen a un estado

**Iniciativa de ley:**

* ID
* Nombre
* Descripción
* Tipo: Local / Federal
* Meta
* Id del estado. Pertenece a un estado si es local
* Tiene muchos apoyos
* Tiene muchas etapas

**Etapas:**

* ID
* Nombre (Recolección de firmas / Presentación de iniciativa / Discusión en cámara de diputados / Discusión en cámara de senadores / Dictamen / Ley Promulgada)
* Id de iniciativa, pertenece a una iniciativa
* Estatus: Pendiente / Terminada / Atorada
* Detalles

**Apoyo para la iniciativa ciudadana de ley:**

* ID
* Número, relativo a la iniciativa ciudadana de ley
* Id de la iniciativa ciudadana de ley / pertenece a una iniciativa ciudadana de ley
* Apellido paterno
* Apellido materno
* Nombre(s)
* CE (Clave de elector)
* OCR
* NE (Número de emisión)
* Firma o huella
* Twitter - \*No requerido
* Id del usuario / candidato | usuario

**Entrada de blog**

* ID
* Fecha de edición
* Título
* Autor
* Id del admin, pertenece a un admin

**Administrador**

* ID
* Tipo de acceso: completo o restringido (Roles)
* **Search for the difference between a GET and a POST request in HTTP. Explain. Show how to handle each type of request in a backend technology.**

**GET**: Requests the data from a specified resource.

* Only limited amount of data can be sent because data is sent in header.
* Is not secured because data is exposed in URL bar.
* Is idempotent, it means second request will be ignored until response of first request is delivered.
* Can be bookmarked.
* More efficient and used more than Post.

**POST**: Submits the processed data to a specified resource.

* Large amount of data can be sent because data is sent in body.
* Is secured because data is not exposed in URL bar.
* Cannot be bookmarked.
* Is non-idempotent.
* Less efficient and used less than get.
* **Explain what is JSON. Explain what is XML. Use your relevant data and convert it into a JSON and XML structure.**

*JSON (JavaScript Object Notation)* is a lightweight data-interchange format. Easy for humans to read and write and also for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.  
JSON is built on two structures:

1. A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.
2. An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

*XML* (Extensible Markup Language) is a markup language similar to HTML. Is a W3C recommended specification as a general purpose markup language. This means, unlike other markup languages, XML is not predefined so you must define your own tags. The primary purpose of the language is the sharing of data across different systems, such as the Internet. *XHTML, MathML, SVG, XUL, XBL, RSS,* and *RDF* are also languages based on XML.

**ALL NEXT POINTS ARE IMPLEMENTED ON LOOPBACK APPLICATION PROJECT AND INDEX HTML FILE:**

* **Select relevant data for the user (at least 4) You are going to use this data to create your web services. Define methods and parameters to consume the data. Define the response.**
* **Search and implement, in the backend technology of your choice, a way to generate a JSON and a XML.**
* **Generate JSON and XML for your data. Use GET parameters to change the data that your are creating. Use meaningful error messages.**
* **Search and implement, in the backend technology of your choice, a way to consume and parse JSON and XML. (From another team)**
* **Use AJAX to consume your data. Document (in html) the examples for the user.**
* **Search for a way to implement cache (backend)**