

TECNICATURA SUPERIOR EN DESARROLLO DE SOFTWARE

DISEÑO WEB

Trabajo Práctico Entregable

Realizar una aplicación web con contenido dinámico en la que se manipulen los registros obtenidos de una aplicación externa (base de datos de estudiantes).

- Se deberá consumir la siguiente API:
[\(url proporcionada por el docente\)](#)
- EndPoints:
 - (GET) student/getAll -> devuelve un listado de todos los estudiantes.
 - (POST) student/ -> agrega un nuevo estudiante.
 - (POST) student/{id}/update -> actualiza un estudiante.
 - (POST) student/{id}/delete -> borra un estudiante.
- Modelo de ejemplo de la estructura JSON:

```
undefined - Users.json
1  [
2    {
3      "userId" : 1,
4      "firstName" : "Denni",
5      "lastName" : "Bevan",
6      "email" : "dbevan0@google.com.hk",
7      "gender" : "Female",
8      "lastConnectedAddress" : "247.25.237.0"
9    }
10 ]
```

- Modelo de ejemplo HTML:

#	First Name	Last Name	Email	Gender
1	Denni	Bevan	dbevan0@google.com.hk	Female
2	Manuel	Birks	mbirks1@arizona.edu	Male
3	Martyn	Flahy	mflahy2@apple.com	Male
4	Camel	Cockhill	ccockhill3@t-online.de	Female
5	Elly	Rontrree	erontrree4@unc.edu	Female
6	Petina	Haithwaite	phaithwaite5@gov.uk	Female
7	Phineas	Balazin	pbalazin6@histats.com	Male
8	Dafy	Neubigging	dneubigging7@desdev.cn	Female
9	Ofelia	Willingam	owillingam8@seattletimes.com	Female
10	Ava	Chessman	achessman9@weather.com	Female

< 1 2 3 4 5 6 7 8 9 10 ... 50 >

- Cada EndPoint realiza una acción distinta. Se deberán crear páginas HTML y JavaScript para:
 1. Obtener y mostrar un listado de todos los estudiantes.
 2. Cargar todos los datos y agregar un nuevo estudiante.
 3. A la opción 3, agregarle la funcionalidad para actualizar los datos de un estudiante.
 4. A la opción 3, agregarle la funcionalidad para dar de baja un estudiante.