



# UNAH

UNIVERSIDAD NACIONAL  
AUTÓNOMA DE HONDURAS

DEPARTAMENTO DE  
INGENIERÍA EN SISTEMAS

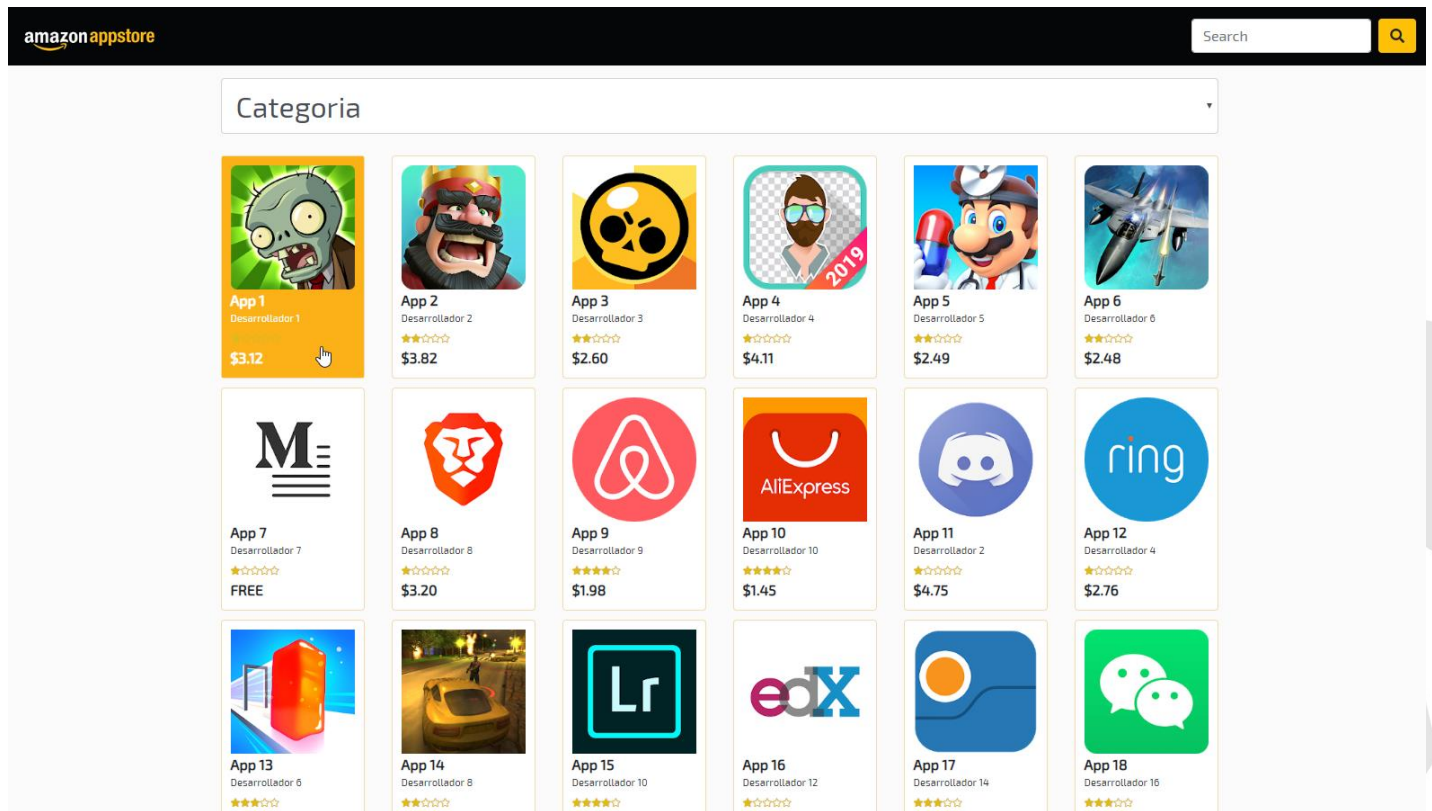
Desarrollo Web  
Examen Práctico Unidad I

Catedrático: Erick Vladimir Marín

“¿Ves aquello que hay más allá del horizonte? ¡Es tu nueva vida sin exámenes! Ya queda poco.” -\_Anónimo

### Diseño:

Diseña un sitio con la apariencia de la imagen llamada sitio.png adjunta a este enunciado. Aquí se muestra en baja resolución. El objetivo es desarrollar un pequeño prototipo frontend de la plataforma Amazon Appstore, la información que se visualiza en la página deberá ser generada a partir de JSONs almacenados en el IndexedDB del navegador. La imagen siguiente no es el sitio original por lo cual se recomienda deducir los estilos a utilizar basados en la imagen expuesta. Para dispositivos grandes y extragrandes se deben visualizar 6 aplicaciones en una fila, para medianos 4 en una fila y para pequeños dos aplicaciones por fila.



Antes de comenzar, se le adjunta un proyecto base el cual incluye carpeta de imágenes, Bootstrap, jQuery y fontawesome. El archivo index.html incluye las etiquetas básicas del ejemplo Starter Template de Bootstrap. Para las estrellas (rellenas y sin relleno) utilizar iconos de fontawesome (Framework ya incluido). Note que para cada aplicación cambia de color en naranja al pasar sobre ella. Todo el sitio debe tener la fuente del siguiente enlace: <https://fonts.google.com/specimen/Exo+2>



También se incluye un archivo javascript con el código necesario para generar la información con la que se trabajará, consiste en una pequeña función para generar información aleatoria para que sea utilizada de base en esta evaluación, **no debe modificar nada de dicha función**, podrá notar que al final queda como resultado un arreglo de JSONs llamado categorías el cual se imprime en la consola y tiene una estructura similar a la siguiente (solo se muestra un fragmento), analícelo detenidamente, se recomienda verlo directamente en la consola del navegador:

```
[
  {
    "nombreCategoria": "Categoria 0",
    "descripcion": "Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolore, modi!",
    "aplicaciones": [
      {
        "codigo": 1,
        "nombre": "App 1",
        "precio": "1.50",
        "descripcion": "Quos numquam neque animi ex facilis nesciunt enim id molestiae.",
        "icono": "img/app-icons/1.webp",
        "instalada": false,
        "app": "app/demo.apk",
        "calificacion": 3,
        "descargas": 1000,
        "desarrollador": "Desarrollador 1",
        "imagenes": [
          "img/app-screenshots/1.webp",
          "img/app-screenshots/2.webp",
          "img/app-screenshots/3.webp"
        ],
        "comentarios": [
          {
            "comentario": "Quos numquam neque animi ex facilis nesciunt enim id molestiae.",
            "calificacion": 2,
            "fecha": "12/12/2012",
            "usuario": "Juan"
          },
          {
            "comentario": "Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolore, modi!",
            "calificacion": 1,
            "fecha": "12/12/2012",
            "usuario": "Pedro"
          },
          {
            "comentario": "Non impedit illum eligendi voluptas. Delectus nisi neque aspernatur asperiores.",
            "calificacion": 4,
            "fecha": "12/12/2012",
            "usuario": "Maria"
          }
        ]
      },
      {
        "codigo": 2,
        "nombre": "App 2",
        "precio": "0.3",
        "descripcion": "Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolore, modi!",
        "icono": "img/app-icons/2.webp",
        "instalada": false,
        "app": "app/demo.apk",
        "calificacion": 4,
        "descargas": 1000,
        "desarrollador": "Desarrollador 2",
        "imagenes": [
          "img/app-screenshots/1.webp",
          "img/app-screenshots/2.webp",
          "img/app-screenshots/3.webp"
        ],
        "comentarios": [
          {
            "comentario": "Quos numquam neque animi ex facilis nesciunt enim id molestiae.",
            "calificacion": 2,
            "fecha": "12/12/2012",
            "usuario": "Juan"
          },
          {
            "comentario": "Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolore, modi!",
            "calificacion": 1,
            "fecha": "12/12/2012",
            "usuario": "Pedro"
          },
          {
            "comentario": "Non impedit illum eligendi voluptas. Delectus nisi neque aspernatur asperiores.",
            "calificacion": 4,
            "fecha": "12/12/2012",
            "usuario": "Maria"
          }
        ]
      }
    ]
  }
]
```



```
"comentarios": [
  {
    "comentario": "Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolore, modi!",
    "calificacion": 2,
    "fecha": "12/12/2012",
    "usuario": "Juan"
  },
  {
    "comentario": "Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolore, modi!",
    "calificacion": 3,
    "fecha": "12/12/2012",
    "usuario": "Pedro"
  },
  {
    "comentario": "Quos numquam neque animi ex facilis nesciunt enim id molestiae.",
    "calificacion": 3,
    "fecha": "12/12/2012",
    "usuario": "Maria"
  }
]
...
}
```

Luego de analizar detenidamente el JSON concluirá que es un arreglo de 5 categorías, cada categoría tiene un atributo llamado aplicaciones el cual es un arreglo que tiene 10 ítems, es decir 10 aplicaciones, cada aplicación tiene un atributo llamado comentarios el cual también es un arreglo de JSONs con la información referente a los comentarios de cada aplicación, a su vez las aplicaciones tienen un atributo llamado imágenes el cual es un arreglo de cadenas para almacenar las URLs de los screenshots de la aplicación.

#### Funcionalidad:

1. La primera vez que cargue la página deberá crear una base de datos llamada **appstore** en IndexedDB, en dicha base de datos crear un **Object Store** llamado **categorías** y llenarlo con la información generada en el arreglo de categorías, deberá almacenar un ítem por cada elemento del arreglo de categorías (es decir en el Object Store deberían quedar 5 ítems). Todas las lecturas de generación de información deberán leer los registros del Object Store.
2. Llenar el select list de categorías con las respectivas almacenadas en el Object Store.
3. Generar el listado de aplicaciones tal y como lo ilustrado en la primera imagen, el listado de las aplicaciones a visualizar debe estar restringido para la categoría seleccionada, para lograr esto deberá registrar en evento onchange del select list para visualizar las aplicaciones correspondientes cada vez que se seleccione una categoría específica. Las estrellas deberán generarse a partir del atributo calificación, utilice las estrellas de fontawesome (cabe recalcar el uso de dos tipos de estrellas una con relleno y otra sin relleno, por ejemplo: si la calificación es 3 se deben dibujar 3 estrellas rellenas y dos sin relleno). En cuanto al precio en caso de ser menor a 0.5 mostrar el texto FREE.

Evento onchange

```
<select id="categoria" onchange="cargarAplicaciones();" >
  ...
</select>
```

```
function cargarAplicaciones(){
  console.log("Cargar aplicaciones de la categoria: " + $("#categoria").val());
}
```

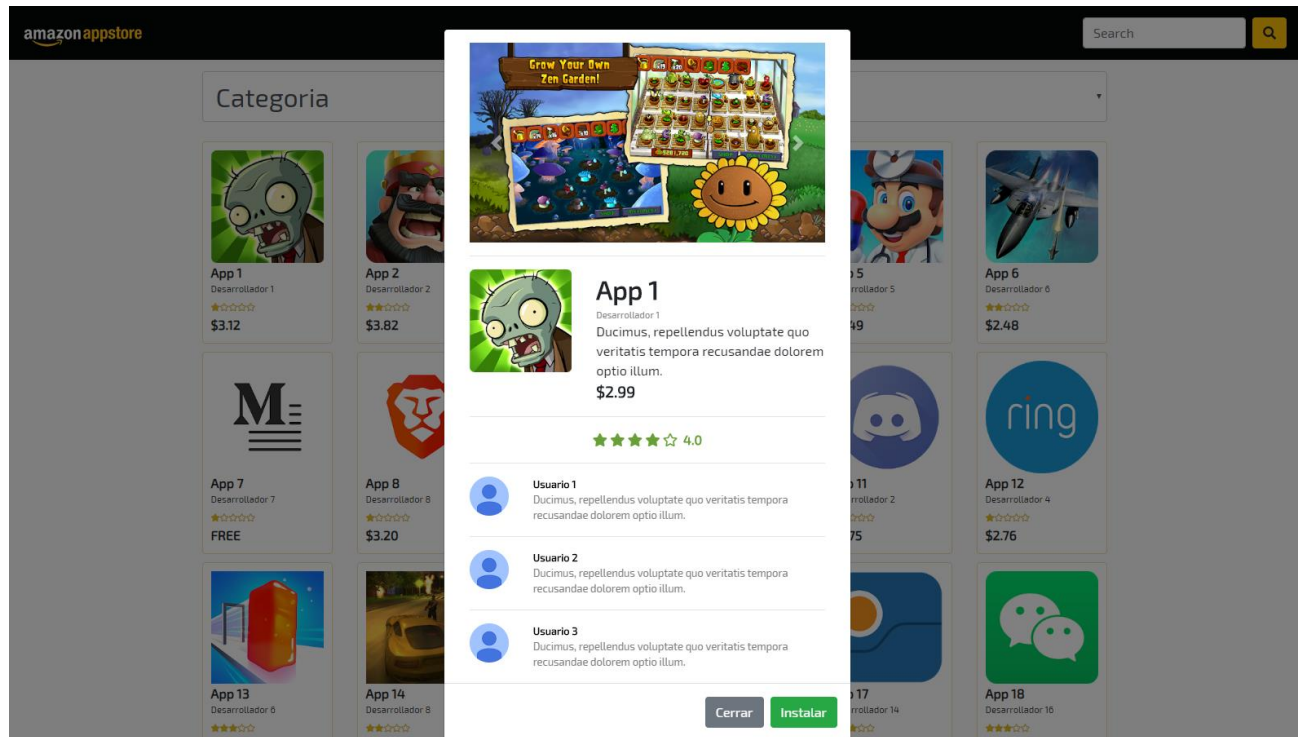
4. Cada vez que se de click sobre una aplicación se deberá abrir una ventana modal mostrando el detalle de la aplicación seleccionada (Cuidado, solo debe existir una ventana modal, el contenido se debe cambiar dependiendo de la aplicación)

seleccionada), para abrir una ventana modal utilizando javascript se hace de la siguiente forma:

```
$('#idVentanaModal').modal('show');
```

Documentación ventanas modales: (<https://getbootstrap.com/docs/4.3/components/modal/>)

El estilo de la ventana modal deberá ser como la mostrada a continuación:



La información a visualizar deberá ser generada dependiendo de la aplicación seleccionada, los comentarios se deberán generar a partir del atributo de comentarios el cual se trata de otro arreglo de JSONs.

Las estrellas deberán de estar en color verde si son tres o más y en color rojo en caso de tener dos o menos.

- En la ventana modal también deberá visualizarse las imágenes que son screenshots de ejemplo, las url de estas imágenes está en un atributo del mismo nombre para cada aplicación, todas las aplicaciones tienen las mismas imágenes según la información almacenada en los JSONs. Las imágenes deberán visualizarse en un carousel de Bootstrap, para utilizar el carousel puede ver la siguiente documentación:  
<https://getbootstrap.com/docs/4.3/components/carousel/#with-controls>
- El botón instalar únicamente se deberá visualizar si el atributo instalado está en false para cada aplicación respectiva. Dicho botón no tendrá funcionalidad alguna.
- Agregar una opción que muestre una ventana modal para agregar una nueva aplicación en la categoría seleccionada, en caso de dicho formulario solo capturar la información básica (para los comentarios e imágenes poner los arreglos vacíos). No es necesario que aplique estilos a dicho formulario.
- Agregar una opción para eliminar una aplicación mediante su código.