

Materia - Evento	Fundamentos de Ingeniería de Software - Obligatorio 2
Código grupo - Docente:	M3B - Martin Solari
Estudiantes del equipo	Alejandro Duran (176652), Santiago Teske (188160)
Fecha de entrega	23/11/2020
URL del repositorio	<a href="https://github.com/ORT-FIS-202008/ob2-duran-teske-1.git">https://github.com/ORT-FIS-202008/ob2-duran-teske-1.git</a>

## Indice

- [Indice](#)
- [Introducción](#)
- [Calidad del código](#)
  - [SonarLint](#)
  - [Indent-Rainbow](#)
  - [Prettier](#)
- [Prueba unitaria](#)
- [Interfaz de usuario](#)
  - [Análisis de fuente](#)
  - [Análisis de botones](#)
  - [Análisis de colores/interfaz](#)
  - [Comparacion de fondo de WhatsApp Web vs el fondo elaborado para el proyecto](#)
- [Construcción](#)
  - [Proceso de BUILD](#)
    - [Ejecución de la aplicación desde los archivos locales](#)
    - [Instalación de Node.js](#)
    - [Ejecución de la aplicación desde el servidor local](#)
    - [Implementación de archivo ejecutable](#)
    - [Implementación de archivo .apk \(Android\)](#)
- [Testing funcional](#)
  - [Casos de prueba](#)
    - [Menú principal: clases de equivalencia](#)
    - [Menú principal: casos de prueba](#)
    - [Nueva Lección : clases de equivalencia](#)
    - [Nueva Lección : casos de prueba](#)
    - [Lecciones: clases de equivalencia](#)
    - [Lecciones: casos de prueba](#)
  - [Prueba Exploratoria](#)
    - [Sesión de testing exploratorio 1](#)
    - [Sesión de testing exploratorio 2](#)
- [Reporte de defectos](#)
  - [Defectos de prioridad baja](#)
    - [ID#4 en Menú Principal se mueve el botón lecciones](#)
    - [ID#5 en Nueva Lección la tablatura permite ingresar caracteres mayores a 24](#)
    - [ID#7 en la versión .apk se muestra alerta de caracter inválido para todos los caracteres](#)
    - [ID#10 Permite registrar una tablatura estando la tablatura vacía](#)
    - [ID#13 Alerta de máximo de caracteres aparece indebidamente](#)
    - [ID#15 Canción e intérprete se pueden guardar con espacios en blanco](#)
    - [ID#16 Si se encoje extremadamente "Nueva Lección" las etiquetas se superponen a los campos de la form](#)
    - [ID#20 El sistema permite ingresar caracteres válidos sin sentido para la tablatura](#)
  - [Defectos de prioridad media](#)
    - [ID#11 Tempo permite ingresar valores fuera de rango](#)
    - [ID#12 Ausencia del mensaje "No hay lecciones creadas"](#)
    - [ID#14 La lista de Canciones no aparece ordenada alfabeticamente](#)
    - [ID#3 Al presionar el borde de una celda no habilita a escribir dentro](#)
  - [Defectos de prioridad alta](#)
    - [ID#18 Al ingresar un caracter en la tablatura el sistema responde "Número máximo de caracteres permitidos es 5"](#)
    - [ID#19 Falla en la reproducción de la tablatura](#)
    - [ID#21 Botón Play no responde ocasionalmente](#)
- [Análisis crítico sobre la calidad del software](#)

## Introducción

Al inicio del curso se nos planteó aprender e implementar distintas técnicas para el desarrollo y producción de software de calidad. Para ello se nos presentó la siguiente problemática : enseñarle a un niño a tocar guitarra. [En aquel entonces](#) nos vimos obligados a abordar una problemática totalmente ajena a lo que

acostumbramos hacer, debimos entonces recurrir a distintas técnicas de la ingeniería de software para poder adaptarnos al problema y posteriormente plantear una solución a él.

Para esta segunda parte del proyecto se nos requirió la implementación de dos funciones que solucionen la problemática planteada al inicio, una para permitir a un profesor de guitarra crear una tablatura y una segunda para permitir al alumno la lectura de dicha tablatura. Dado que el enfoque de este obligatorio es el aprendizaje e implementación de técnicas de ingeniería de software que garanticen un producto de calidad debimos cambiar nuestras aspiraciones iniciales (aquellas planteadas en el obligatorio anterior) dado que los tiempos de implementación son cortos en relación a lo necesario para poder llegar al producto que pensábamos inicialmente con la calidad esperada para este segundo obligatorio.

Es así que para este segundo obligatorio nos propusimos implementar una función "Nueva Lección" que permitiera la creación de una tablatura de guitarra, en la que se incluirían todos los símbolos técnicos propios de una tablatura de guitarra, el nombre de la canción, nombre de intérprete, dificultad y tempo. Para la segunda función, "Lecciones" tomaríamos una tablatura creada previamente para poder reproducirla a una velocidad acorde a la canción original o leerla de forma estática si el usuario así lo deseara.

Dadas estas funcionalidades a lo largo del segundo obligatorio se hizo énfasis en generar dos funciones simples que cumplieran con las características del software de calidad. Fue así que buscamos la implementación de una buena Interfaz de Usuario, el desarrollo de código de calidad, asegurar la realización de testing mediante las técnicas adecuadas y un posterior reporte de defectos que contuviera aquellos defectos encontrados en la fase posterior al desarrollo.

El resultado fue una aplicación capaz de ser utilizada por personas que sepan leer y escribir tablaturas, de manera sencilla y rápida. Puede ser utilizada con fines educativos o recreativos. De todas formas, vale destacar que nuestra aplicación está dirigida a niños.

## Calidad del código

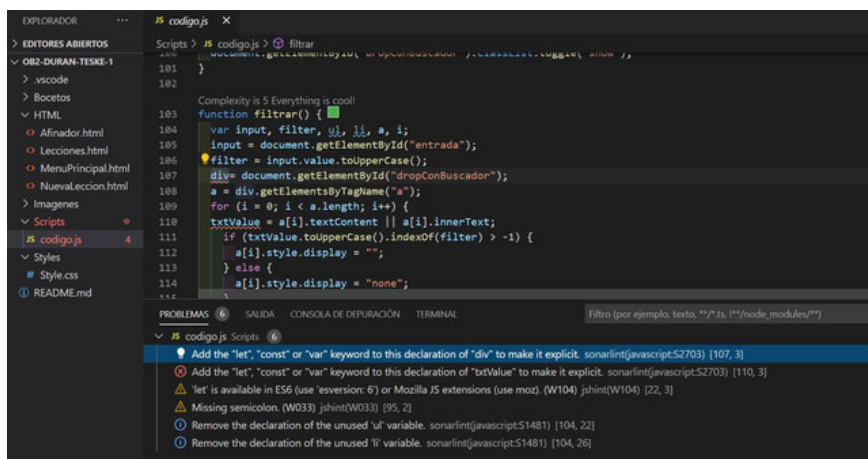
Para el desarrollo de este proyecto nos basamos en las guías de estilo de Google, tanto para [JavaScript](#) como para [HTML/CSS](#). Consideramos que fue una buena práctica el adaptar estas guías de estilo ya que estableció un estándar para la codificación, de manera de lograr que ambos miembros del equipo generaran código similar. Previamente a la selección de estas guías de estilo ya se había comenzado con el desarrollo de código fuente, por lo que fue necesario implementar cambios en:

- Indentación.
- Control de mayúsculas y minúsculas.
- Remoción de los atributos de tipo en las etiquetas.
- Remoción de ceros a la izquierda antes de valores, por ejemplo: 0,5.
- Se añadió "-" (guion) entre los identificadores del selector. Por ejemplo: ".primerEjemplo" -> ".primer-ejemplo".
- Se ordenó alfabéticamente las declaraciones o propiedades dentro del archivo .css.
- Corrección de separación de llaves "{" en las declaraciones.
- Separar declaraciones por 2 espacios (enters).

Aplicar buenas prácticas de codificación. Además, para asegurar una mejor calidad del código y evitar errores durante la fase de desarrollo utilizamos herramientas a modo de extensiones de Visual Studio tales como SonarLint, ESLint e Indent-Rainbow. Mientras las primeras dos herramientas se encargaron de la detección de errores (por ejemplo) del tipo sintáctico o semántico, mientras que Indent-Rainbow nos ayudó a tener un mejor control sobre la indentación de nuestro código. Finalmente utilizamos también la herramienta Prettier, que fue de gran ayuda a la hora de darle formato al código, no dejando lugar a errores.

Algunos ejemplos de uso de las herramientas:

### SonarLint



### Indent-Rainbow

```

13 <div class="div-general-nueva-leccion">
14 <h2>Nueva Leccion</h2>
15 <div class="div-superior-nueva-leccion">
16 <div class="div-tabla-nueva-leccion">
17 <table class="tabla-superior-nueva-leccion">
18 <tr>
19 <td style="width: 20%;>CANCION</td>
20 <td><input type="text" name="NombreCancion"></td>
21 </tr>
22 <tr>
23 <td style="width: 20%;>INTERPRETE</td>
24 <td><input type="text" name="Interprete"> </td>
25 </tr>
26 <tr>
27 <td style="width: 20%;>DIFICULTAD </td>
28 <td><select name="OpcionesDificultad" class="selector-dificultad">
29 <option value="1" selected>Facil</option>
30 <option value="2" >Intermedio</option>
31 <option value="3">Avanzado</option>
32 </select></td>
33 </tr>
34 </table>
35 </div>
36 </div>
37 <div>
38 <button class="boton-guardar">Guardar</button>
39 <button onclick="abrirMenu()" class="boton-atras"></button>
40 </div>
41 </div>
42 </body>

```

## Prettier

```

66 Complety is 3 Everything is cool
67 function mostrarCancion() {
68   var datosRecuperados = localStorage.getItem("lecciones");
69   var arrayLecciones = JSON.parse(datosRecuperados);
70   var nombreCancion = this.textContent;
71   for (var i = 0; i < arrayLecciones.length; i++) {
72     if (nombreCancion == arrayLecciones[i].cancion) {
73       var labelDificultad = document.getElementById("label-dificultad");
74       var labelCancion = document.getElementById("label-cancion");
75       var labelInterprete = document.getElementById("label-interprete");
76       var labelTempo = document.getElementById("label-tempo-valor");
77       labelDificultad.textContent = "Dificultad: " + arrayLecciones[i].dificultad;
78       labelCancion.textContent = "Cancion: " + arrayLecciones[i].cancion;
79       labelInterprete.textContent = "Interprete: " + arrayLecciones[i].interprete;
80       labelTempo.textContent = arrayLecciones[i].tempo;
81     }
82   }
83   dibujarTabla(arrayLecciones[i].ObjTabla);
84 }
85 }
86 }
87 }
88 }

```

A la izquierda podemos ver el código previo al uso de la herramienta Prettier, mientras que la derecha tenemos el código resultado.

## Prueba unitaria

Implementamos pruebas automatizadas con el objetivo de detectar fallas en el software, evitando que el programador tenga que ejecutar las pruebas manualmente. Para la automatización de las pruebas unitarias utilizamos Jest que es un framework de pruebas para JavaScript. Lo primero a realizar fue la selección de funciones que queríamos probar y considerábamos por el tiempo disponible del proyecto mas importantes para testear. Creamos la clase *NuevaLeccion.test.js* donde incluimos nuestros casos de pruebas. Las pruebas se realizaron desde la terminal de VSCode arrojando los siguientes resultados finales

```

PASS Jest/NuevaLeccion.test.js
✓ Corroborar que todos los campos esten completos (6 ms)
✓ Corroborar que si todos los campos estan incompletos falle (1 ms)
✓ Corroborar que si algun campo estan incompleto falle
✓ Corroborar que si algun campo estan incompleto falle (1 ms)
✓ Corroborar que si algun campo estan incompleto falle
✓ Corroborar que si algun campo estan incompleto falle (1 ms)
✓ Corroborar que si algun campo estan incompleto falle
✓ Corroborar que si algun campo estan incompleto falle (1 ms)
✓ Corroborar que si algun campo estan incompleto falle
✓ Corroborar que si algun campo estan incompleto falle (1 ms)
✓ Corroborar que sean 5 los caracteres ingresados
✓ Corroborar que sean 5 los caracteres ingresados (1 ms)
✓ Corroborar que sean 5 los caracteres ingresados
✓ Corroborar que sean 5 los caracteres ingresados (1 ms)
✓ Corroborar que sean 5 los caracteres ingresados
✓ Corroborar que sean 5 los caracteres ingresados (1 ms)
✓ Valida que las teclas son: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que las teclas son: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que las teclas son: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que las teclas son: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que la tecla son: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que la tecla son: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que la tecla son: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que la tecla son: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que la tecla son: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que la tecla pertenezca a: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48 (1 ms)
✓ Valida que la tecla pertenezca a: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ tecla no es ninguna de estas: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48 (1 ms)
✓ tecla no es ninguna de estas: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48 (1 ms)
✓ tecla no es ninguna de estas: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ tecla no es ninguna de estas: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48 (1 ms)
✓ tecla no es ninguna de estas: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ tecla no es ninguna de estas: b,h,p,h,s,49,50,51,52,53,54,55,56,57,48
✓ Valida que este la cancion en la lista de canciones
✓ Valida que no este la cancion en la lista de canciones (1 ms)
✓ Valida que no este la cancion en la lista de canciones
✓ Valida que este la cancion en la lista de canciones (1 ms)

Test Suites: 1 passed, 1 total
Tests: 38 passed, 38 total
Snapshots: 0 total
Time: 4.712 s
Ran all test suites.

```

Con el comando `npm run test-coverage` nos devolvió la siguiente tabla estadística.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	15.24	50	30.77	15.15	
Leccion.js	16.67	100	0	16.67	2-6
NuevaLeccion.js	15.15	50	33.33	15.05	3-132

Test Suites: 1 passed, 1 total  
 Tests: 38 passed, 38 total  
 Snapshots: 0 total  
 Time: 8.655 s  
 Ran all test suites.

Nos pareció una muy buena herramienta para evitar hacer pruebas manuales que pueden dejar casos de prueba fuera y ganar tiempo, la parte negativa de Jest en comparación con otras herramientas es que no permite automatizar browsers.

## Interfaz de usuario

Para la realización de la interfaz de usuario nos basamos en varios conceptos del concepto de [Material Design](#) de Google. Durante la elaboración del proyecto recurrimos a distintos recursos externos así como [fuentes de texto](#) e [íconos](#) que se adaptaran a dicha filosofía. Por otro lado utilizamos como inspiración software que sabemos que cumple los estándares de calidad que pretendemos alcanzar, un ejemplo de esto se puede ver en el fondo elegido para nuestra aplicación, inspirado en el fondo de WhatsApp Web. Para la elaboración de dicho fondo utilizamos elementos relacionados al mundo de la música, la guitarra y el rock, intentando siempre crear algo amigable para los usuarios objetivos, los niños. Para ello procedimos a la utilización de emojis de guitarra y música, llegando a un fondo de tema "oscuro" pero al mismo tiempo colorido.

### Análisis de fuente

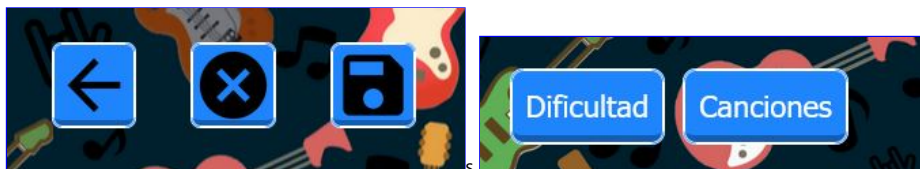
Durante la implementación del sistema utilizamos la fuente recomendada por Google "Roboto- Sans Serif" en diversos tamaños. Decidimos hacer uso de esta fuente para todos los elementos de la interfaz, utilizando diferenciación por tamaños para los elementos que queríamos destacar. Otro factor que nos pareció importante a la hora de decidir utilizar esta fuente es que es la recomendada para dispositivos Android.

### Análisis de botones

Siguiendo el estandar especificado de Material Design de Google implementamos los botones de acuerdo a nuestra necesidad. Seguimos estándares de:

- tamaño
- colores
- formas
- márgenes
- bordes
- posicionamiento
- simplicidad de íconos

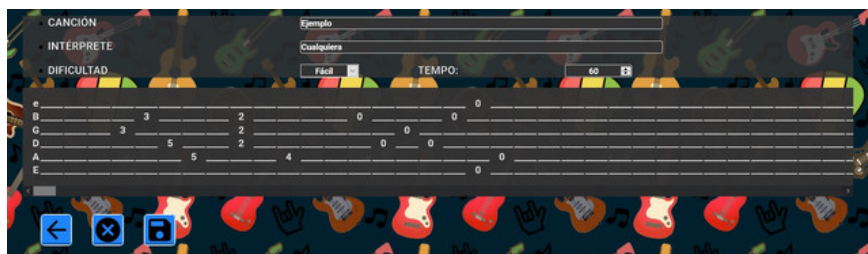
Durante el proceso de diseño de los botones tuvimos en cuenta los ítems anteriormente mencionados. Es decir, buscamos que nuestros botones tuvieran una correlación con el fondo en el que estaban dispuestos en cuanto a su color, de manera de que fueran resaltables pero al mismo tiempo estuvieran en armonía con la interfaz en general. Por otro lado establecimos una única forma para nuestros botones que se repite en todo el sistema, generando una sensación de uniformidad. En cuanto al contenido de nuestros botones decidimos utilizar íconos de la galería de íconos de google que ya cumplen los requisitos buscados por defecto (imágenes en dos dimensiones, fáciles de reconocer, orgánicos, etc).



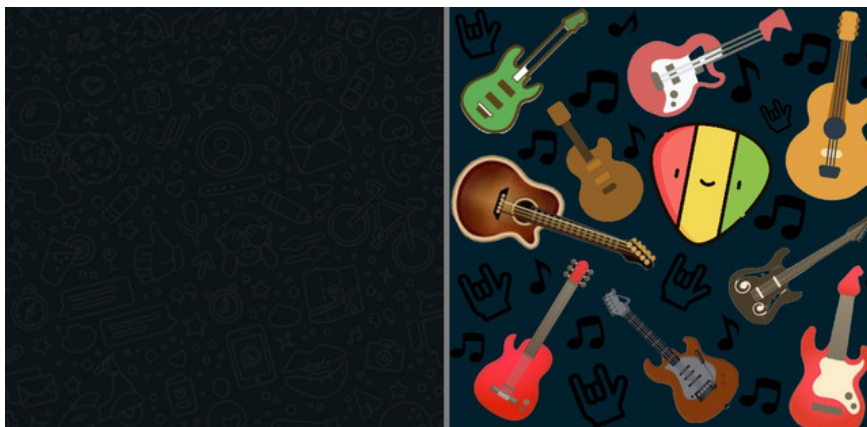
### Análisis de colores/interfaz

Basamos nuestra selección de colores en el fondo elegido para la aplicación. Es decir, buscamos en principio algo que recuerde a las interfaces "Dark Theme" actuales y que al mismo tiempo resultara llamativo a los niños. Fue así que manejamos una gama de colores que va desde el gris claro, pasando por azul oscuro y tonos de celeste, hasta llegar al blanco (para las fuentes). Utilizamos marcos cuadrados para destacar el área de ingreso de datos y al mismo tiempo recurrimos a formas con bordes curvos para los elementos de campo de texto o ingreso de datos. Otra decisión de diseño en este sentido fue la implementación de un ligero grado de transparencia en los elementos principales del sistema, lo que nos permitió enfatizar aquello que necesitábamos en cada elemento.





Comparacion de fondo de WhatsApp Web vs el fondo elaborado para el proyecto



---

## Construcción

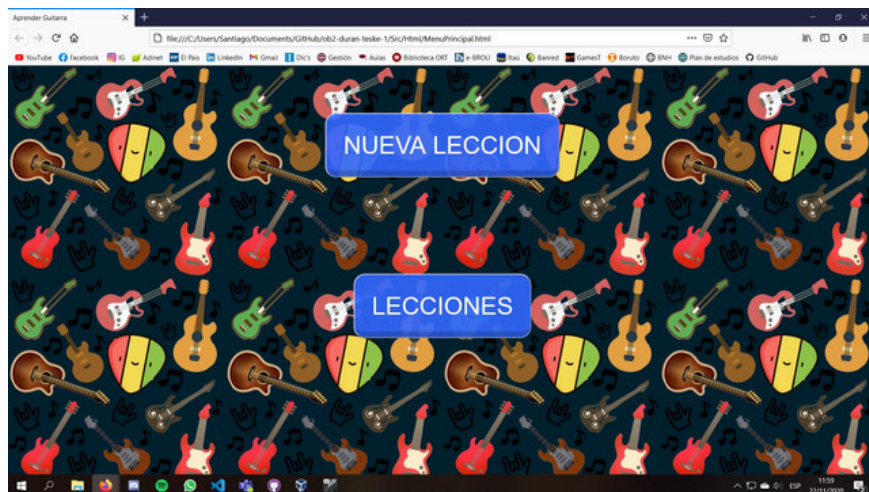
Como objetivo de este proyecto se nos solicitó la implementación de dos funciones con el fin de brindar una lección de guitarra a un niño. Estas funciones son "Nueva Lección" para la creación de nuevas lecciones y "Lecciones", desde donde el niño podrá acceder a un listado de lecciones creadas previamente, seleccionar una y proceder a la visualización de la tablatura de dicha canción.

Para la creación de dicha aplicación fueron utilizadas las tecnologías HTML5, CSS3, Javascript y Node.js. El resultado fue una aplicación web adaptable a distintos dispositivos.

### Proceso de BUILD

Ejecución de la aplicación desde los archivos locales

Ya que nuestra aplicación es del tipo web vale destacar que la forma mas sencilla de ejecutarla sería abrir directamente su archivo principal (.html) de forma manual a través del navegador web. Si bien durante esta sección exploraremos otras formas de ejecutar la aplicación creemos necesario recordar que siempre existe esta posibilidad. Para ejecutar la aplicación manualmente debemos acceder a la ruta donde se encuentra el repositorio, posteriormente acceder a la carpeta "**Src**", luego ingresar a la carpeta "**Html**" y finalmente abrir el archivo "**MenuPrincipal.html**". Algo que debemos destacar es que **no será posible** utilizar esta metodología **desde el navegador Safari** si usted es usuario de **MacOs**. En ese caso le pedimos que intente ejecutar la aplicación con Chrome o Firefox desde su ordenador o siga los pasos citados [aquí](#).



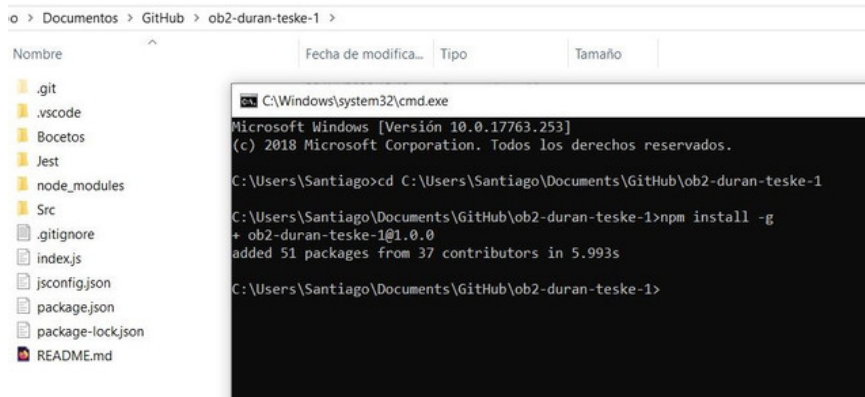
Ejemplo utilizando Firefox en Windows10

Instalación de Node.js

Tanto para el desarrollo de la aplicación, como para su testeo o implementación hicimos uso de distintas herramientas de Node.js. Es decir, para ejecutar la aplicación será necesario que instale Node.js y algunos de sus módulos los cuales citaremos a continuación. Pasos a seguir :

1. Descargar Node.js desde su [sitio oficial](#). Recomendamos la versión LTS.
2. Seguir los pasos de instalación.
3. Recomendamos reiniciar el equipo luego de la instalación.
4. Una vez finalizados estos pasos debe abrir la consola de comandos de su sistema operativo, posicionarse en la carpeta del repositorio y ejecutar el comando:

**npm install -g.**

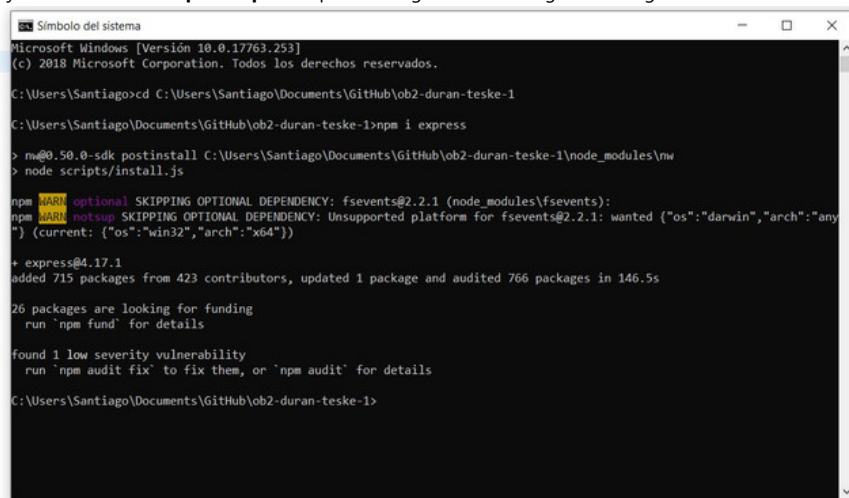


Una vez finalizado este paso podemos continuar con el proceso de BUILD.

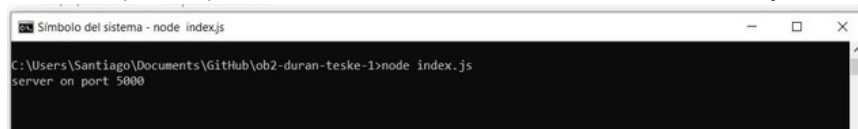
### Ejecución de la aplicación desde el servidor local

Una vez instalado Node.js podemos proceder a la instalación de nuevos módulos para la implementación de un servidor local, para poder de esa forma correr nuestra aplicación en él. Para ello debemos:

1. Abrir la consola de comandos
2. Posicionarnos en la carpeta del repositorio
3. Ejecutar el comando **npm i express**. Aparecerá algo similar a la siguiente imagen.

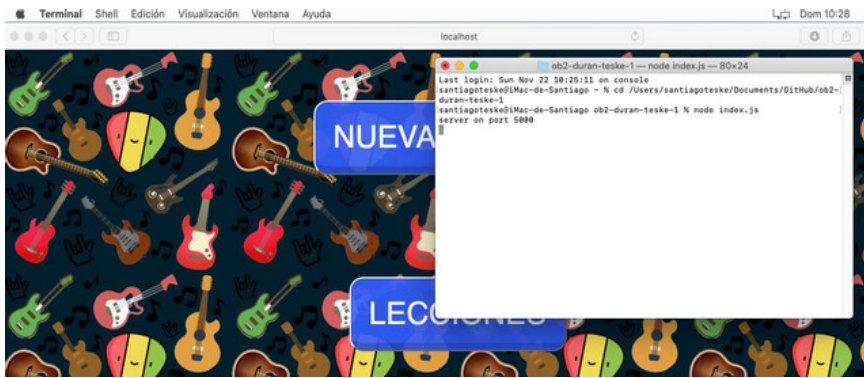


4. Una vez finalizada la instalación podemos proceder a abrir el servidor local a través del comando **node index.js**, debemos obtener una respuesta similar a la



siguiente imagen.

5. Una vez finalizado este paso debemos abrir nuestro navegador web y escribir en la barra de direcciones **localhost:5000**. Esto nos llevará directamente a la aplicación web. Veremos como ejemplo la ejecución de este comando en MacOS.

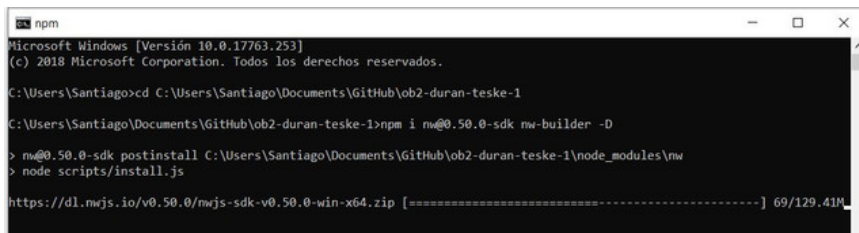


Vale destacar que de esta forma los usuarios de MacOS podrán acceder directamente a la aplicación web a través de Safari sin la necesidad de utilizar Chrome o Mozilla Firefox. Otro punto que consideramos necesario aclarar es que para la instalación de los módulos necesarios se necesita contar con permisos de administrador y permitir el acceso de fuentes externas al ordenador, de otra forma no será posible la instalación de los `node_modules` necesarios para la implementación del servidor local. Queremos recordar que en la rama "Ejecutable" de nuestro repositorio se encuentra una copia de los módulos necesarios (bajo el nombre `node_modules`) para la implementación en caso de haber problemas con la descarga.

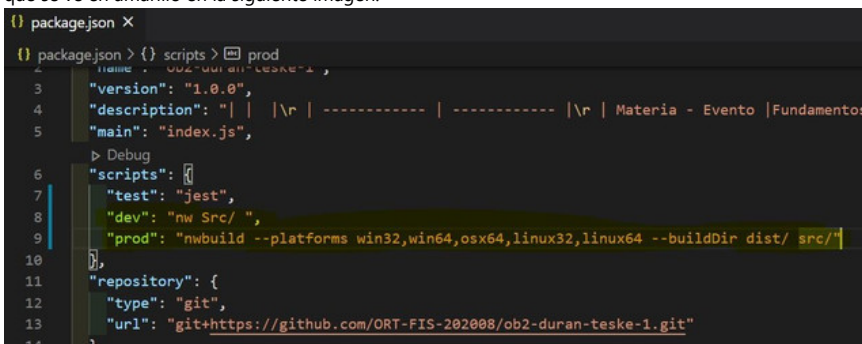
### Implementación de archivo ejecutable

Otra forma de correr nuestra aplicación será a través de un archivo ejecutable. Para la creación de dicho archivo ejecutable nos basamos en el uso de [NW.js](#). Esto nos permitirá tomar nuestro proyecto y transformarlo en un ejecutable capaz de abrir la aplicación fuera del navegador web. Para la creación de dicho ejecutable debemos:

1. Abrir la consola/terminal de nuestro sistema.
2. Posicionarnos en la carpeta raíz de nuestro proyecto.
3. Ejecutar el comando **`npm i nw@0.50.0-sdk nw-builder -D`** como se puede apreciar en la siguiente imagen.



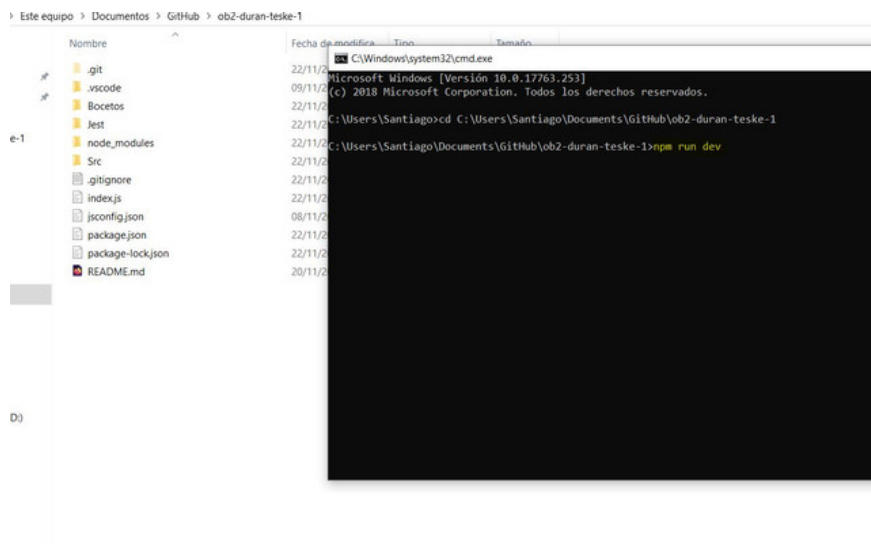
4. Una vez finalizado dicho proceso debemos ir a la carpeta del repositorio y abrir el archivo `package.json`. Una vez ahí debemos corroborar la línea `"prod"= ...` que se ve en amarillo en la siguiente imagen.



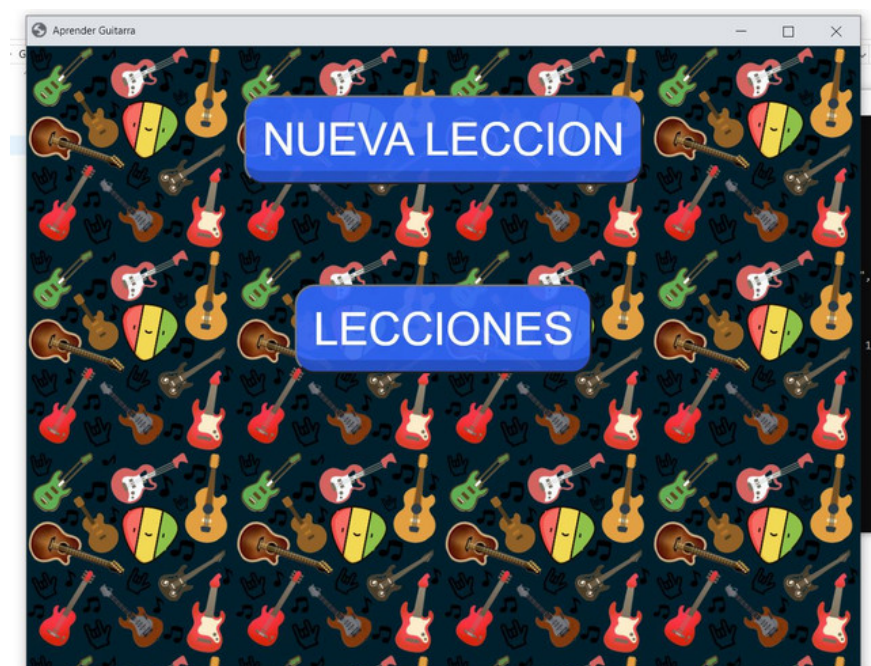
Esta línea indicará en qué sistemas operativos serán ejecutados nuestros archivos ejecutables. Por defecto hemos dejado win32 y win64 dado que son los sistemas más utilizados. De todas maneras, es posible agregar ejecutables para MacOS y Linux a través de esta línea si copiamos el ejemplo en amarillo de la imagen anterior. *Nota: hasta el momento no hemos podido implementar un ejecutable para el sistema MacOS (osx64), nos hemos percatado de que este error le sucede a múltiples usuarios, por lo que recomendamos que si se está utilizando una Mac se recurra a los tipos de implementación anteriores.*

5. Luego de chequear dicho archivo debemos solamente volver a la consola, posicionarnos nuevamente en la raíz del proyecto y escribir el código **`npm run dev`**, esto ejecutará el programa como se ve en la siguiente imagen, aunque **no supone** la construcción del archivo ejecutable para su distribución aún.





*Ejecutable*



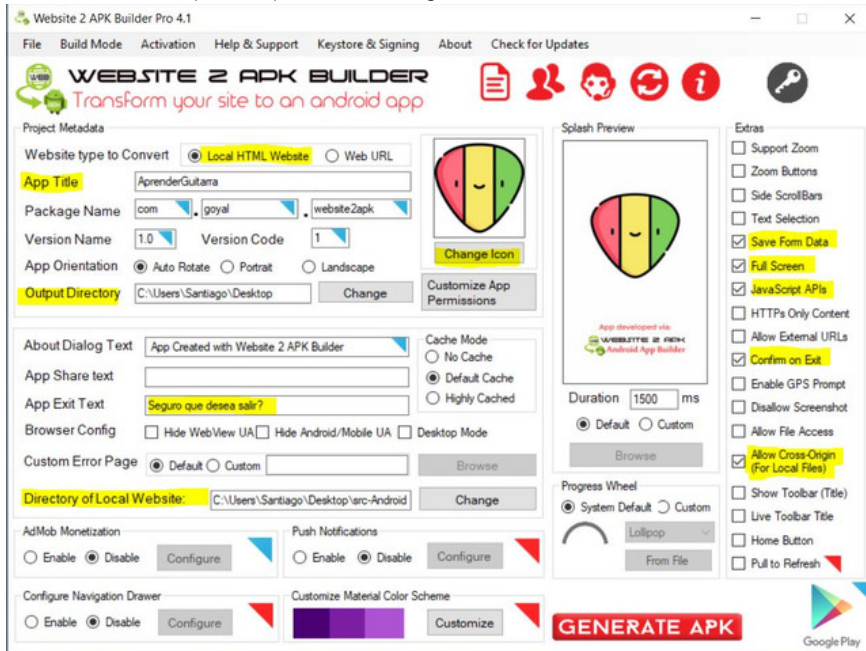
6. Finalmente procederemos a la implementación del/los archivos ejecutables. Para ello sencillamente se debe volver a la consola en el directorio raíz del proyecto y ejecutar el comando **npm run prod**. Esto generará una carpeta de nombre **"dist"** dentro del directorio raíz del proyecto y dentro de ella encontraremos distintas carpetas con una instalación distinta para cada sistema operativo que hayamos elegido en el archivo package.json .
7. Finalmente cliqueamos en el archivo **"AprenderGuitarra.exe"** y se ejecutará la aplicación.





Finalmente, de manera de probar nuestra aplicación en teléfonos celulares decidimos buscar una forma de convertir nuestro proyecto a una aplicación para Android en formato .apk. Para lograr dicho objetivo nos valimos de una aplicación externa de nombre [Website 2 APK Builder](#). Si bien el uso de esta aplicación es extremadamente sencillo fue necesaria la re-estructuración del proyecto en cuanto a la organización de sus carpetas para la implementación del .apk. Gracias a ello decidimos agregar la carpeta con el proyecto pronto para su conversión dentro del repositorio, dicha carpeta se puede encontrar en el repositorio como **"/android/src-Android"**. Pasos para la implementación del .apk :

1. Descargar la aplicación [Website 2 APK Builder](#).
2. Ejecutar su instalador y seguir los pasos de instalación por default.
3. Ejecutar la aplicación.
4. Una vez dentro de la aplicación podremos ver la siguiente ventana:



Para facilitar el instructivo hemos pintado de amarillo los campos a tener en cuenta. Solo será necesario configurar los campos de igual manera que en la imagen anterior a excepción de "Output Directory" (a gusto del usuario) y "Directory of Local Website" (debe ser la carpeta **src-Android** presente en el repositorio). Una vez completado este paso debemos clicar "GENERATE APK".

5. Una vez finalizado el proceso deberíamos tener un archivo .apk en la ruta de salida seleccionada anteriormente. A partir de ese punto solo queda correr la aplicación en cualquier dispositivo Android. *Nota: en la primera ejecución el programa solicitará permisos para leer/escribir carpetas, compartir ubicación o usar cámara y video. Recomendamos denegar todas estas peticiones.*

## Testing funcional

Para la realización de estas pruebas nos basamos en el modelo de caja negra, particularmente en el modelo de clases de equivalencia, donde establecimos cual era el comportamiento esperado del software en los escenarios mencionados a continuación. Para complementar la etapa de testing decidimos que era necesario realizar también sesiones de testing exploratorio a modo de encontrar nuevos defectos que hubieran pasado desapercibidos hasta el momento.

### Casos de prueba

Menú principal: clases de equivalencia

Entrada/ variable	Clases válidas	Clases inválidas
Nueva Lección	Se cliquea el borde del botón "Nueva Lección" y se ingresa a la ventana "Nueva Lección" (1)	- Se cliquea "Lecciones" por error (5)
	Se cliquea el interior del botón "Nueva Lección" y se ingresa a la ventana "Nueva Lección" (2)	- Se cliquea fuera de la opción "Nueva Lección" (6)
Lecciones	Se cliquea el borde del botón "Lecciones" y se ingresa a la ventana "Lecciones" (3)	- Se cliquea "Nueva Lección" por error (7)
	Se cliquea el interior del botón "Lecciones" y se ingresa a la ventana "Lecciones" (4)	- Se cliquea fuera de la opción "Lecciones" (8)

Menú principal: casos de prueba

Caso de prueba	Click en	Resultado Esperado	Clases de equivalencia cubiertas
CP1	Interior del botón Nueva Lección	Se ingresa a la ventana "Nueva Lección"	2

Caso de prueba	Click en	Resultado Esperado	Clases de equivalencia cubiertas
CP2	Borde del botón Nueva Lección	No se produce ningún cambio	1
CP3	Un punto aleatorio de la pantalla	No se produce ningún cambio	6,8
CP4	Interior del botón Lecciones	Se accede a la ventana "Lecciones"	4
CP5	Borde del botón Lecciones	Se accede a la ventana "Lecciones"	3
CP6	Borde del botón Nueva Lección, por error	Se ingresa a Nueva Lección	7
CP7	Interior del botón Lecciones, por error	Se ingresa a Lecciones	5

#### Nueva Lección : clases de equivalencia

Entrada/ variable	Clases válidas	Clases inválidas
Canción	Canción no registrada previamente (1)	Canción ya registrada (11)
		Campo vacío (12)
Intérprete	Se completa el campo con el nombre del intérprete (2)	Se deja el campo vacío (13)
Dificultad	Selección de una de las tres categorías válidas (3)	
Tempo	Número mayor/igual a 30 y menor/igual a 210 (4)	Campo vacío (14)
		Número menor a 30 o mayor a 210 (15)
		Caracteres no numéricos (16)
Tablatura	Celda con 5 o menos caracteres (5)	Más de 5 caracteres (17)
	Celda con número mayor/igual a cero y menor/igual a 24 (6)	Caracter no válido (18)
	Celda de formato número-carácter-número, donde carácter es p,s,/,h,b,v o x (7)	Número mayor a 24 (19)
Botón Guardar	Click en el botón, guarda la lección (8)	Faltan elementos en los campos Canción, Intérprete, Dificultad, Tempo o se está frente a una tabla vacía (20)
		El usuario presiona fuera del botón (21)
Botón Reestablecer	Click en el botón, elimina toda la información ingresada previamente (9)	El usuario presiona fuera del botón (22)
Botón volver	Click en el botón, se regresa al menú principal.(10)	El usuario presiona fuera del botón (23)

#### Nueva Lección : casos de prueba

Casos de Prueba	Canción	Intérprete	Dificultad	Tempo	Tablatura	Guardar	Reestablecer	Volver	Resultado esperado	Clases de equivalencias cubiertas
CP1	American Idiot	Green Day	Intermedio	150	Números entre 0-24, caracteres : x,s,v	Click en interior de Guardar	N/A	N/A	El sistema toma la tablatura y muestra en pantalla "Creada con éxito"	1,2,3,4,5,6,7,8
CP2	<b>American Idiot</b>	Green Day	Fácil	170	Números entre 0-24, caracteres : x,s,v	Click en interior de Guardar	N/A	N/A	Mensaje : "Canción registrada previamente"	<b>11</b> ,2,3,4,5,6,7
CP3	Otherside	<b>vacío</b>	Fácil	100	Números entre 0-15, caracteres: s	Click en borde de Guardar	N/A	N/A	Mensaje "Ingresa el nombre del Intérprete"	1, <b>13</b> ,3,4,5,6,7, <b>20</b>

Casos de Prueba	Canción	Intérprete	Dificultad	Tempo	Tablatura	Guardar	Reestablecer	Volver	Resultado esperado	Clases de equivalencias cubiertas
CP4	Enter Sandman	Metallica	Difícil	29	Números entre 0-24, caracteres: b,p,h,v,/,x	Click en interior de Guardar	N/A	N/A	Mensaje "El Tempo debe ser mayor/igual a 30 y menor/igual a 210"	1,2,3,15,5,6,7
CP5	Blitzkrieg Bop	Ramones	Fácil	vacío	Números entre 1-12, caracter: x	Click en borde del botón "Guardar"	N/A	N/A	Mensaje: "El campo "Tempo" no puede estar vacío"	1,2,3,14,5,6,7,20
CP6	Eye of the Tiger	Survivor	Fácil	moderado	Números entre 1-10	N/A	N/A	N/A	Mensaje : "El campo Tempo solo acepta números"	1,2,3,16,5,6
CP7	Hotel California	The Eagles	Difícil	80	Se ingresa: 12b144	N/A	N/A	N/A	Mensaje: "Máximo de caracteres permitidos por nota es 5"	1,2,3,4,17,19
CP8	Message in a Bottle	The Police	Intermedio	110	Se ingresa: 5r	N/A	N/A	N/A	El sistema no permite toma el caracter "r"	1,2,3,4,18
CP9	Hold the Line	Toto	Intermedio	80	Se ingresa el número 134	N/A	N/A	N/A	Mensaje: "Inserte un número entre 0 y 24"	1,2,3,4,19
CP10	A las 9	NTVG	Facil	30	Se ingresan números entre 1-10	Click en el borde de botón Guardar	N/A	Click en el interior del botón "Volver"	Mensaje: "Creada con éxito", luego de aceptar el mensaje regresa al menú principal.	1,2,3,4,5,6,7,8,10
CP11	Back in Black	AC/DC	Intermedio	vacío	Click en interior de botón Guardar	N/A	N/A	Mensaje: "Por favor, complete la tablatura."	1,2,3,4,20	
CP12	Sweet Child O'Mine	Guns and Roses	Difícil	211	Números entre 0-24, caracteres: b,p,h,v,/,x	Click en interior de Guardar	N/A	N/A	Mensaje "El Tempo debe ser mayor/igual a 30 y menor/igual a 210"	1,2,3,15,5,6,7
CP13	Aint' Talkin Bout Love	Van Halen	Difícil	210	Números entre 0-24, caracteres: b,p,h,v,/,x	Click en Guardar	N/A	Click en el borde del botón "Volver"	Mensaje "Creada con éxito", luego de aceptar el mensaje regresa al menú principal	1,2,3,4,5,6,7,8,10

Casos de Prueba	Canción	Intérprete	Dificultad	Tempo	Tablatura	Guardar	Reestablecer	Volver	Resultado esperado	Clases de equivalencias cubiertas
CP14	Californication	Red Hot Chili Peppers	Facil	80	Números entre 0-24	N/A	Click en el interior de "Restablecer"	Click en "Volver"	Se eliminan todos los datos introducidos y vuelve al menú principal.	1,2,3,4,5,6,7,9,10
CP15	Californication	Red Hot Chili Peppers	Facil	80	Números entre 0-24	El usuario presiona fuera del botón Guardar	N/A	N/A	No sucede nada	1,2,3,4,5,6,7,21
CP16	Island on the Sun	Weezer	Facil	85	Números entre 0 y 12	N/A	El usuario presiona fuera del botón Restablecer	N/A	No se borran los datos de la cancion	1,2,3,4,5,6,7,22
CP17	Super Freak	Rick James	Facil	90	Números entre 0 y 17, caracteres	N/A	N/A	El usuario presiona fuera del botón Volver	No sucede nada	1,2,3,4,5,6,7,23

Lecciones: clases de equivalencia

Entrada/ variable	Clases válidas	Clases inválidas
Dificultad	Se cliquea el interior del botón Dificultad y se despliegan sus opciones (1)	Se cliquea fuera de la opción Dificultad(13)
	Se cliquea el borde del botón Dificultad y se despliegan sus opciones (2)	
Canciones	Se cliquea el interior del botón Canciones y se despliegan sus opciones (3)	Se cliquea fuera de la opcion Canciones (14)
	Se cliquea el borde del botón Canciones y se despliegan sus opciones (4)	
Inicio	Se cliquea el interior del botón Inicio (5)	Se cliquea fuera del botón inicio (15)
	Se cliquea el borde del botón Inicio (6)	
Play	Se cliquea el interior del boton Play (7)	Se cliquea fuera del botón Play (16)
	Se cliquea el borde del botón Play(8)	
Pausa	Se cliquea el interior del botón Pausa (9)	Se cliquea fuera del botón Pausa (17)
	Se cliquea el borde del botón Pausa (10)	
Fin	Se cliquea el interior del botón Fin(11)	Se cliquea fuera de Fin (18)
	Se cliquea el borde del botón Fin (12)	
Difícil	Se cliquea la opción Difícil dentro del menú Dificultad (19)	Se cliquea fuera de la opción Difícil (24)
Intermedio	Se cliquea la opción Intermedio dentro del menú Dificultad (20)	Se cliquea fuera de la opción Intermedio (25)
Fácil	Se cliquea la opción Fácil dentro del menú Dificultad (21)	Se cliquea fuera de la opción Fácil (26)
Todas	Se cliquea la opcion Todas dentro del menú Dificultad (22)	Se cliquea fuera de la opción Todas (27)
Cancion	Se selecciona una cancion del menú Canciones(23)	Se cliquea fuera de la opción Cancion(28)

Lecciones: casos de prueba

Caso de prueba	Click en	Resultado Esperado	Clases de equivalencia cubiertas
CP1	El interior de Canciones, sin haber cliqueado Dificultad antes.	Se despliega la lista de todas las canciones.	3
CP2	El interior de Canciones, habiendo cliqueado Dificultad antes.	Se despliega la lista de canciones filtradas por dificultad	3



Caso de prueba	Click en	Resultado Esperado	Clases de equivalencia cubiertas
CP3	El borde de Canciones, sin haber clickeado Dificultad antes.	Se despliega la lista de todas las canciones.	4
CP4	El borde de Canciones, habiendo clickeado Dificultad antes.	Se despliega la lista de canciones filtradas por dificultad	4
CP5	Fuera del botón Canciones, por error.	No sucede nada	18
CP6	El interior de Dificultad	Se despliega la lista de dificultades	1
CP7	El borde de Dificultad	Se despliega la lista de dificultades	2
CP8	Fuera del botón Dificultad, por error.	No sucede nada.	17
CP9	El interior del botón Play, posicionado en el inicio de la tablatura.	Comienza la reproducción de la tablatura .	7
CP10	El interior del botón Play, posicionado en el fin de la tablatura.	No sucede nada.	7
CP11	El interior del botón Play, durante la reproducción	No suced nada.	7
CP12	El borde del botón Play, posicionado en el inicio de la tablatura.	Comienza la reproducción de la tablatura .	8
CP13	El borde del botón Play, posicionado en el fin de la tablatura.	No sucede nada.	8
CP14	El borde del botón Play, durante la reproducción.	No sucede nada.	8
CP15	El exterior del botón Play, por error.	No sucede nada.	16
CP16	El interior del botón Pausa, durante la reproducción.	Se detiene la reproducción de la tablatura.	9
CP17	El interior del botón Pausa, previo a la reproducción.	No sucede nada.	9
CP18	El borde del botón Pausa, durante la reproducción.	Se detiene la reproducción de la tablatura.	10
CP19	El borde del botón Pausa, previo a la reproducción.	No sucede nada.	10
CP20	El exterior del botón Pausa, por error.	No sucede nada.	17
CP21	El interior del botón Inicio, posicionado en el inicio de la tablatura	No sucede nada.	5
CP22	El interior del botón Inicio, posicionado en un punto distinto al inicio.	Transporta al usuario al inicio de la tablatura.	5
CP23	El borde del botón Inicio, posicionado en el inicio de la tablatura	No sucede nada.	6
CP24	El borde del botón Inicio, posicionado en un punto distinto al inicio.	Transporta al usuario al inicio de la tablatura.	6
CP25	El exterior del botón Inicio, por error.	No sucede nada.	19
CP26	El interior del botón Fin, posicionado en el fin de la tablatura	No sucede nada.	11
CP27	El interior del botón Fin, posicionado en un punto distinto al fin.	Transporta al usuario al fin de la tablatura.	11
CP28	El borde del botón Fin, posicionado en el fin de la tablatura	No sucede nada.	12
CP29	El borde del botón Fin, posicionado en un punto distinto al fin.	Transporta al usuario al fin de la tablatura.	12
CP30	El exterior del botón Fin, por error.	No sucede nada.	18
CP31	Opción Difícil	Muestra las canciones de dificultad Difícil en el menu Canciones.	19
CP32	Exterior de opción Difícil, por error.	No sucede nada.	24
CP33	Opción Intermedio	Muestra las canciones de dificultad Intermedio en el menú Canciones.	20
CP34	Exterior de opción Intermedio, por error.	No sucede nada.	25
CP35	Opción Fácil	Muestra las canciones de dificultad fácil.	21

Caso de prueba	Click en	Resultado Esperado	Clases de equivalencia cubiertas
CP36	Exterior de opción Fácil, por error.	No sucede nada.	26
CP37	Opción Todas.	Muestra todas las canciones disponibles.	22
CP38	Exterior de opción Todas,por error.	No sucede nada.	27
CP39	Cancion del menú Canciones.	Carga la canción seleccionada al reproductor de tablaturas.	23
CP40	Exterior del item Cancion en menú Canciones,por error.	No sucede nada.	28

## Prueba Exploratoria

Para obtener un panorama mas amplio acerca del estado de calidad de nuestra aplicación decidimos realizar dos sesiones de prueba exploratoria con el objetivo de reportar posibles fallas o defectos del sistema a mejorar en el futuro.

### Sesión de testing exploratorio 1

Mision	Obtener un panorama general sobre la situación de la aplicación en cuanto a defectos.
Inicio	18:02 22/11/2020 Duración: corta (30 min)
Tester	Alejandro Durán
Estructura de división del tiempo	1.Diseño y ejecución de pruebas 60%
	2. Investigación y reporte de defectos 40%
	3. Configuración de la sesión :0%
	4. Objetivo vs Oportunidad 100%/0%
Notas de pruebas	Ejecuto la aplicación desde localhost:5000
	Ingreso a "Nueva Lección"
	<b>P1:</b> Validación de tablaturas
	Permite guardar una canción sin notas registradas en la tablatura
	DEFECTO: ID 10- Reportado en Issues de GitHub
	<b>P2:</b> Validación del campo Tempo
	Permite ingresar el número 5000 como tempo válido
	DEFECTO: ID 11- Reportado en Issues de GitHub
	<b>P3:</b> Validaciones de texto en general
	La palabra éxito lleva tilde.
	La palabra máximo lleva tilde.
	DEFECTO: <i>corregido en el momento</i> , no se subio a GitHub
	<b>P4:</b> Prueba en menú de dificultad y canciones en "Lecciones"
	Se selecciona Intermedio, no hay canciones cargadas en esa dificultad y posteriormente al seleccionar "Canciones" no aparece ningún mensaje de error.
	DEFECTO : ID 12- Reportado en Issues de GitHub
	<b>P5:</b> Validación de caracteres en la tablatura
	Se ingresa un caracter incorrecto y posteriormente el sistema indica que el máximo de caracteres es 5.
	DEFECTO : ID 13 Reportado en GitHub
	<b>P6:</b> Observación en la lista de canciones de Lecciones.
	La lista de canciones no está ordenada alfabéticamente.
	DEFECTO : ID 14 Reportado en GitHub

### Sesión de testing exploratorio 2

Misión	Hallar defectos que hayan pasado desapercibidos hasta el momento.
Inicio	18:15 22/11/2020 Duración: corta (30 min)
Tester	Santiago Teske
Estructura de división del tiempo	1.Diseño y ejecución de pruebas 70%
	2. Investigación y reporte de defectos 30%
	3. Configuración de la sesión :0%
	4. Objetivo vs Oportunidad 100%/0%
Notas de pruebas	Ejecuto la aplicación desde AprenderGuitarra.exe (win64)
	<b>P1:</b> Validación de datos en Nueva Leccion
	Se puede crear una canción aunque en los campos Canción o Intérprete haya espacios vacíos. " "
	DEFECTO: ID 15 Reportado en GitHub
	<b>P2:</b> Validación de responsividad del sistema
	Si encoge la ventana a niveles extremos se puede observar una deformación en la distribución de los elementos.
	DEFECTO: ID 16 Reportado en GitHub.
	<b>P3:</b> Validación de entradas en la tablatura
	Permite ingresar números de 5 cifras que no representan un traste de guitarra en la realidad.
	DEFECTO: ID 17 Reportado en GitHub
	<b>P4:</b> Validación de ingreso de datos en la tablatura
	Si se intenta ingresar un número muy rapidamente el sistema tiende a mostrar el alert "No se permiten mas de 5 caracteres" más allá que no se haya ingresado nada aún.
	DEFECTO: ID 18 Reportado en GitHub .
	<b>P5:</b> Observación del reproductor de canciones.
	Si estamos reproduciendo una tablatura y decidimos cambiar de canción, la nueva canción seguirá reproduciendose a la velocidad de la primera.
	DEFECTO: ID 19 Reportado en GitHub.

## Reporte de defectos

En esta sección del documento haremos una puesta a punto de los principales defectos detectados durante los procesos de testing. En la sección de Issues de GitHub se encuentra el reporte total de defectos, aquellos que pudieron ser solucionados y aquellos que no. En esta sección nos enfocaremos solamente en aquellos que aún no hayan sido solucionados por distintos motivos.

Para el reporte de defectos en esta documentación dividimos los resultados obtenidos en tres categorías por prioridad: Alta, Media y Baja. Decidimos agregar también un nivel de severidad para cada defecto basado en una escala similar a la anteriormente mencionada. Como los defectos a continuación ya fueron reportados en [GitHub](#) creemos conveniente realizar un breve recuento en esta sección con las características : Prioridad, Severidad y Resumen (de la situación).

### Defectos de prioridad baja

ID#4 en Menú Principal se mueve el botón lecciones

Resumen: Al pasar el mouse por la opción "Nueva Lección" genera el movimiento de dicho botón y al mismo tiempo del botón "Lecciones". Severidad: Baja

ID#5 en Nueva Lección la tablatura permite ingresar caracteres mayores a 24

Resumen: Al crear una nueva tablatura, esta nos permite ingresar valores numéricos fuera de rango. Severidad: Media

ID#7 en la versión .apk se muestra alerta de caracter inválido para todos los caracteres

Resumen: Al crear una tablatura en la versión .apk figura la alerta "Caracter inválido" ante cualquier tipo de entrada que se intente. Severidad: Alta

ID#10 Permite registrar una tablatura estando la tablatura vacía

Resumen: Si en la ventana "Nueva Lección" se completan los datos de "Canción", "Intérprete" y "Tempo" el sistema permitirá crear una tablatura más allá de que esta se encuentre vacía. Severidad: Baja

#### ID#13 Alerta de máximo de caracteres aparece indebidamente

Resúmen: al ingresar un caracter inválido en la sección "Nueva Lección" recibimos la alerta "Máximo de caracteres es 5" Severidad:Alta

#### ID#15 Canción e intérprete se pueden guardar con espacios en blanco

Resúmen: en Nueva Lección se permite ingresar como nombre de canción y/o intérprete un espacio en blanco. Severidad:Baja

#### ID#16 Si se encoje extremadamente "Nueva Lección" las etiquetas se superponen a los campos de la form

Resúmen: Reduciendo el tamaño de la ventana "Nueva Lección" extremadamente se produce la superposición de algunos de sus elementos. Severidad:Alta

#### ID#20 El sistema permite ingresar caracteres válidos sin sentido para la tablatura

Resúmen: El sistema permite el ingreso de caracteres válidos que no reflejan un significado en el lenguaje de la guitarra, por ejemplo "bsx". Severidad: Baja.

### Defectos de prioridad media

#### ID#11 Tempo permite ingresar valores fuera de rango

Resúmen: dentro de "Nueva Lección" la caja "Tempo" permite ingresar válores fuera del rango estipulado. Severidad: Baja

#### ID#12 Ausencia del mensaje "No hay lecciones creadas"

Resúmen: Cuando se selecciona un nivel de dificultad en el cual no hay ninguna cancion creada, el sistema no informa del hecho. Severidad: media.

#### ID#14 La lista de Canciones no aparece ordenada alfabeticamente

Resúmen: al seleccionar una tablatura en el menú Canciones, estas no se muestran ordenadas en orden alfabético. Severidad: Baja

#### ID#3 Al presionar el borde de una celda no habilita a escribir dentro

Resúmen: Al intentar ingresar un caracter cualquier celda de la tablatura, si tocamos el borde de la misma no nos permitira ingresar el dato. Severidad: Media

### Defectos de prioridad alta

#### ID#18 Al ingresar un caracter en la tablatura el sistema responde "Número máximo de caracteres permitidos es 5"

Resúmen: Al intentar ingresar un caracter válido en la sección de tablatura de manera rápida el sistema responde con el alert "Máximo de caracteres permitidos es 5" Severidad: Alta

#### ID#19 Falla en la reproducción de la tablatura

Resúmen: Si mientras estamos reproduciendo una tablatura decidimos abrir una segunda (sin detener la reproducción de la primera) la segunda tomará los valores de Tempo de la primera. Severidad: Alta

#### ID#21 Botón Play no responde ocasionalmente

Resúmen: Si durante la reproducción de una tablatura nos dirigimos al final de esta o la tablatura llega a su fin por naturaleza y posteriormente seleccionamos otra cancion, la segunda canción no iniciará hasta que toquemos Pausa y luego Play nuevamente. Severidad:Alta

## Análisis crítico sobre la calidad del software

---

Considerando los puntos anteriores y teniendo en cuenta a la aplicación en sí hemos determinado las siguientes conclusiones acerca del producto final.

En cuanto a la calidad de interfaz de usuario:

- Sostenemos que la interfaz de usuario es atractiva, que tiene en consideración al usuario objetivo que son los niños, respetando los estándares de calidad de interfaz de usuario de Google.
- Hemos verificado que la aplicación cumple con los criterios necesarios para ser llamada "responsive", ya que fuimos capaces de probarla tanto en el emulador de tamaño de los navegadores como en distintos teléfonos celulares comprobando que la interfaz no se ve modificada y se adapta a cualquier entorno.
- Otro punto a destacar es que la aplicación es intuitiva para el usuario, ya que mantuvimos siempre el concepto de sencillez y a esto le sumamos el uso de íconos fáciles de interpretar.

En cuanto a calidad de código:

- Creemos que el código resultado es de calidad, ya que utilizamos distintos tipos de herramientas y analizadores para conseguir tener un código limpio y fácil de mantener.
- Apoyamos también el punto anterior en que además de las herramientas utilizadas se fijó un estándar de calidad para la tarea de desarrollo y como extra se realizó la tarea de refactorización al final de la fase de desarrollo.
- Como punto negativo o a mejorar consideramos que el código está falto de comentarios y esto podría dificultar su entendimiento por parte de programadores externos.



En cuanto a testing:

- Hallamos útil la implementación de distintas técnicas de testing ya que nos ayudaron a encontrar defectos que no habíamos detectado durante la etapa de desarrollo.
- Al mismo tiempo, consideramos que al proyecto en general le faltó mas contenido de testing. Por lo que es un punto que consideraríamos para futuras versiones.

Otros detalles a mejorar:

- Creemos que sería necesario implementar un botón de ayuda con información acerca de qué caracteres son soportados por la tablatura.
- Sería necesario implementar una versión App. para dispositivos IOs.
- Solucionar las fallas reportadas en la sección de defectos, sobre todos aquellas de prioridad media/alta.