Programación Distribuida Momento Valorativo #2

|  |  |
| --- | --- |
| Docente | Juan Carlos Zuluaga |
| Valor | 20% |
| Fecha de entrega | segunda clase de la semana del 18 de marzo del 2018 |
| Tema | Consumo de una API Rest Full, Xamarin Forms, Patrones: Locator & MVVM, Manejo multi lenguaje |
| Condiciones | * Trabajo individual * La nota depende de la sustentación * Si se encuentra copia parcial o total entre trabajos, la nota será de 0.0 para todos los implicados * En caso de ser un grupo numeroso, el docente, a su discreción y criterio, decide quienes sustentan y quienes no * Todo el software, comentarios, commits debe ser en idioma inglés. * La interfaz de usuario debe estar disponible en Ingles (idioma por defecto) y debe tener otros 4 idiomas, en Español más otros 3 que usted decida (francés, alemán, italiano, holandés, griego, cualquier idioma que usted escoja). * Debe utilizar el patrón MVVM para la App. Recuerde nuestra campaña #NoAlCodeBehind * Su código debe ser ordenado, claro y debe seguir todas las buenas prácticas de programación que hemos visto hasta el momento en clase * Toda fuente debe residir en el repositorio GitLab y debe observarse un progreso cronológico en los Commits realizados. * No se reciben trabajos que no sean entregados en el CD tal y como se describe más adelante en este documento. |

# Enunciado

# Vamos a construir una App de una Biblia. Vamos a basarnos en el API <http://api.biblesupersearch.com> y en el API que construimos en clase para la App de “Lands” para todo lo que tiene que ver con la autenticación. En líneas generales la App debe tener la siguiente funcionalidad:

* La aplicación debe mostrarse en 5 idiomas diferentes. Inglés como idioma predeterminado, más español, más otros 3 idiomas que usted seleccione. Esto es en relación a las etiquetas, títulos y mensajes de la interfaz de usuario. No tiene nada que ver con el idioma que seleccione el usuario de la biblia que se seleccione. Es decir, si el dispositivo está configurado en Ingles, se debe mostrar la App en Ingles, o en el idioma que esté el dispositivo. Si el dispositivo está configurado en un idioma diferente a los 5 idiomas que usted implemente, se deberá mostrar en el idioma por defecto, es decir inglés.
* Debe poder ingresar a la App ingresando un email y contraseña. Utilizando el mismo API de Lands.
* Debe poder registrar un nuevo usuario. Utilizando el mismo API de Lands.
* Debe poder ingresar por 2 redes sociales diferentes.
* Debe tener la opción de recordarme en el dispositivo y de recuperar contraseña. Utilizando el mismo API de Lands.
* Luego de la autenticación debe mostrar la lista de biblias disponibles que se obtendrán con el siguiente servicio: <http://api.biblesupersearch.com/api/bibles> por ejemplo:



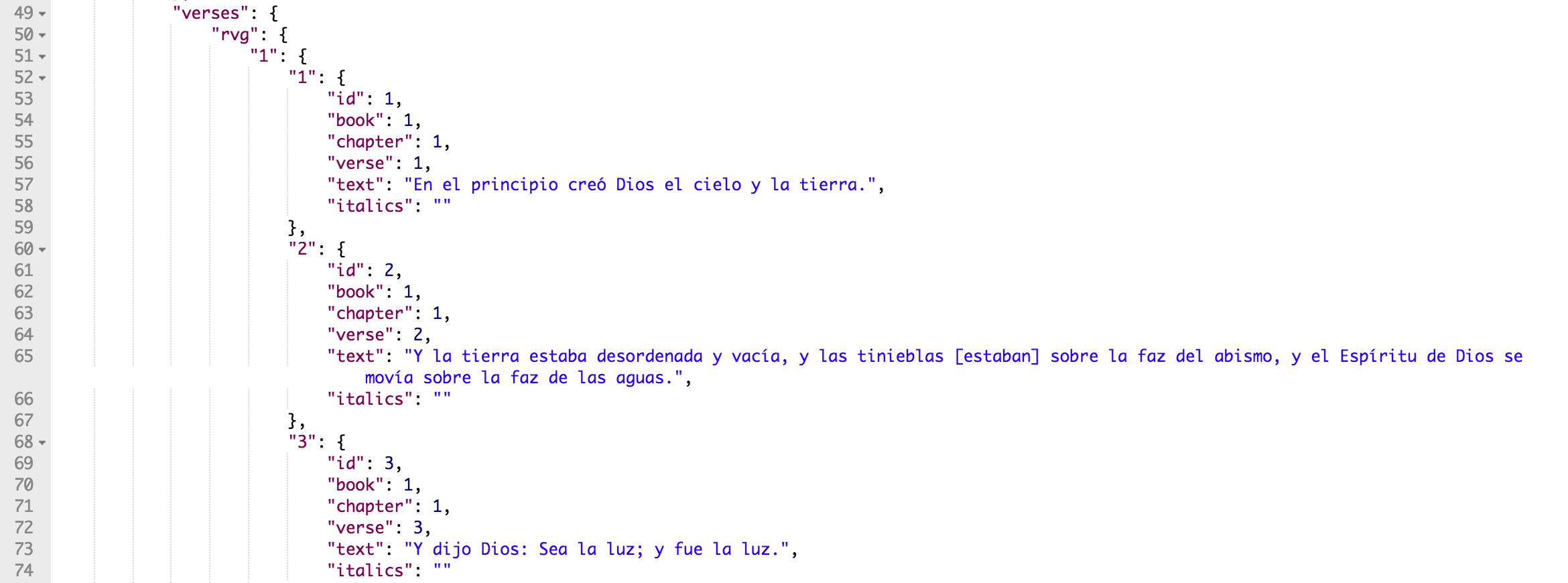
Debe mostrar el nombre de la biblia, el idioma y el año de publicación de la misma en un ListView.

* Luego cuando el usuario seleccione una de las biblias debe mostrar la lista de libros que componen la biblia y estos se deben mostrar en el idioma seleccionado de la biblia, por ejemplo:

|  |  |
| --- | --- |
| <http://api.biblesupersearch.com/api/books?language=en> | <http://api.biblesupersearch.com/api/books?language=es> |
|  |  |

Tenga en cuente que la abreviatura del libro o shortname, que lo vamos a necesitar para la siguiente consulta, solo viene en idioma inglés, para los otros idiomas viene en blanco. Por eso siempre debe obtener la lista de libros en inglés (para obtener las abreviaturas) y en el idioma seleccionado de la biblia para mostrarlo de forma correcta.

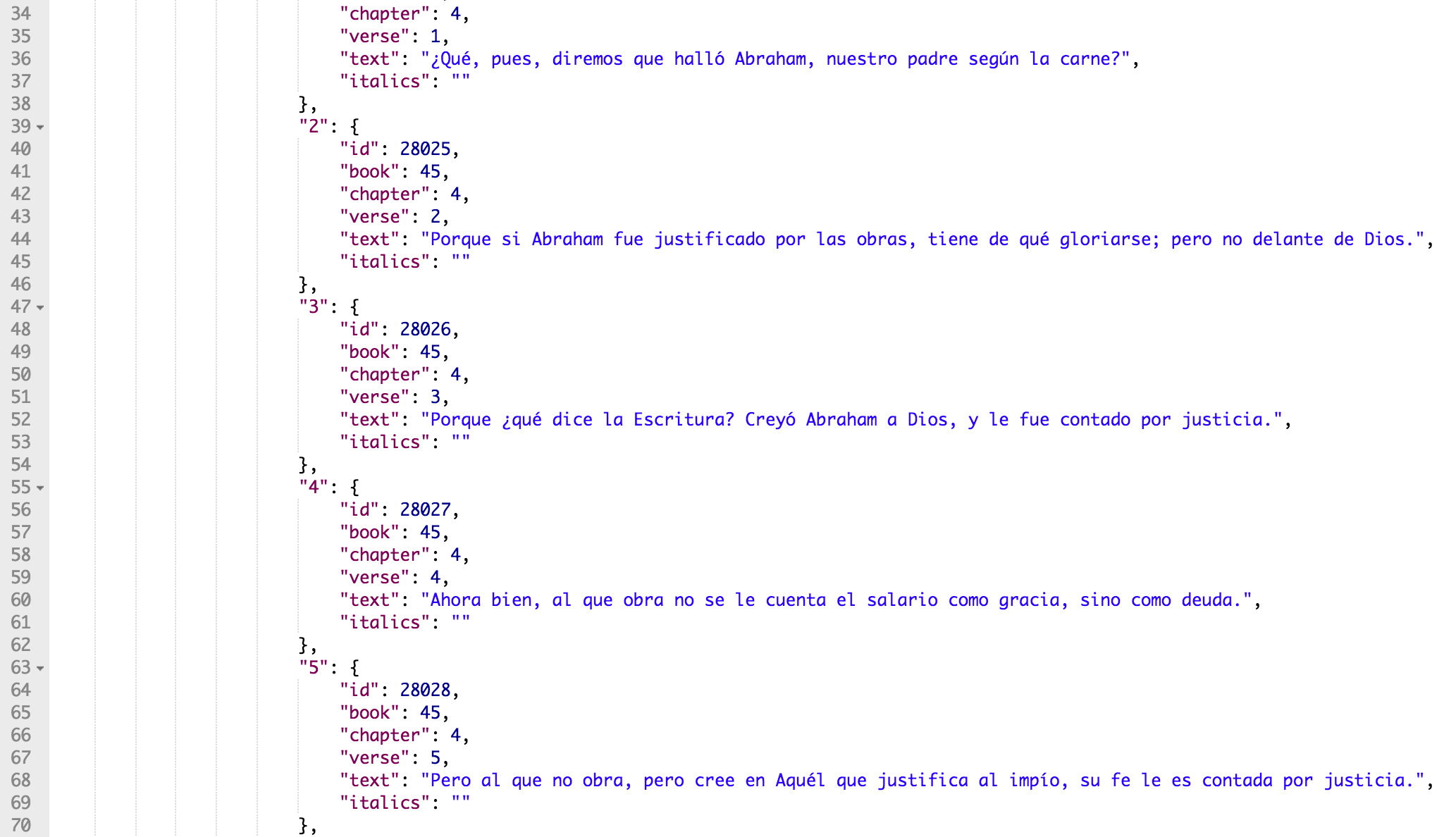
* Luego de seleccionar el libro, debe mostrar el contenido del libro, para eso utilizamos el método: <http://api.biblesupersearch.com/api?bible=rvg&reference=Gen>, donde **bible=rvg** es el código de la biblia de la primera pantalla y **reference=Gen** es la abreviatura del libro en ingles de la segunda pantalla. Con esto podemos mostrar el primer capítulo del libro:



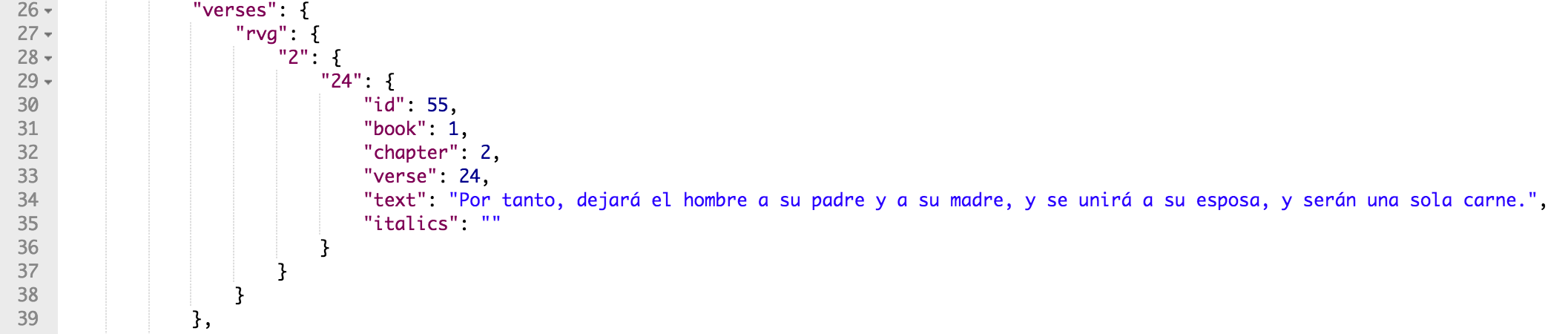
Al final debe haber un botón de “Siguiente” para ver el siguiente capítulo del libro. Para esto te puedes apoyar de las propiedades el objeto que dicen cómo se llama el siguiente capítulo. A partir del segundo capítulo se debe activar el botón “Anterior” para ver el anterior capitulo. Y en el último capítulo no debe estar disponible el botón “Siguiente”.



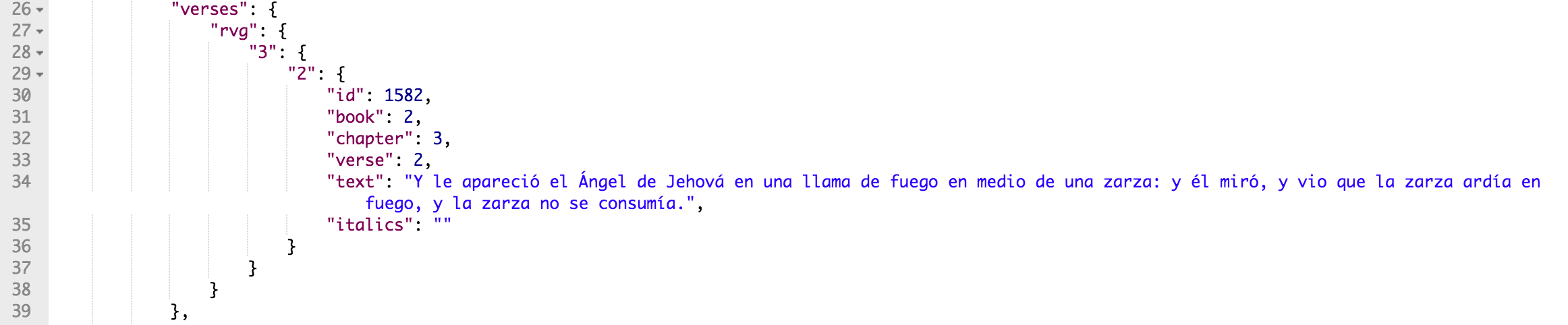
* En el menú hamburguesa de la App, debe estar la opción de Salir, para cerrar la sesión actual y volver a pedir datos de login.
* En el menú hamburguesa de la App, debe tener una opción de búsqueda por citas, donde se debe poder seleccionar el libro, luego escribir el capítulo y el verso o los versos que se desean ver. Por ejemplo si queremos buscar, en la biblia rvg (Reina Valera Gómez), el libro de los Romanos, capitulo 4, versículos 1 al 10, se debe consumir el API de esta manera: https://api.biblesupersearch.com/api?bible=rvg&reference=Rom 4:1-10



* En el menú hamburguesa de la App, debe tener una opción de búsqueda por palabra clave, donde se debe poder seleccionar el libro (opcional) e ingresar la palabra o frase de búsqueda por ejemplo: “Amor” y se deben mostrar todos los versos donde se encuentra la palabra clave en el libro seleccionado o en todos los libros. Por ejemplo si queremos buscar la palabra “Padre”, en la biblia RVG en todos los libros, usamos: http://api.biblesupersearch.com/api?bible=rvg&search= Padre



Si queremos buscar la palabra “Ángel”, en la biblia RVG en el libro del Éxodo, usamos: http://api.biblesupersearch.com/api?bible=rvg&search=ángel&reference=Exo



* La App debe ser bonita, agradable, utilizar íconos, gráficos y debe ser intuitiva en su manejo.

# Entregable

El día de la sustentación debe entregar un CD debidamente marcado con su nombre completo y grupo con las siguientes carpetas:

* **Application**: donde va a colocar el proyecto completo de Xamarin.
* **Documentation**: dónde vas a colocar un documento Word con los siguientes títulos:
  + **Portada:** marcas tu trabajo
  + **User:** haces un manual de usuario sencillo con la explicación de cómo funciona tu App con todos los pantallazos.
  + **Goals:** haces una tabla indicando lo que lograste hacer y lo que no lograste hacer de tu trabajo.
  + **Repository:** colocar la URL donde puedo acceder a tu repositorio en GitLab (no olvides agregarme como Developer o Master en tu respositorio)
  + **History:** vas a poner un documento con “pantallazos” donde se vea toda la historia de los commits realizados

# Calificación

|  |  |
| --- | --- |
| **Funcionalidad** | **Puntos** |
| Autenticación local. | 5 |
| Registrar nuevos usuarios. | 5 |
| Registro por 2 redes sociales. | 5 |
| Listar biblias. | 15 |
| Listar libros de la biblia. | 15 |
| Mostrar contenidos de la biblia. | 15 |
| Búsqueda por libro, capitulo, verso. | 15 |
| Búsqueda por palabras clave. | 15 |
| Interfaz de usuario. | 10 |
| **TOTAL** | **100** |

# Recomendaciones

* Empiece hoy mismo
* Su trabajo debe ser UNICO, trabaje individualmente.

¡Mucha suerte!