

10 mo
CONGRESO NACIONAL

CoNaIISI

Congreso Nacional de Ingeniería Informática / Sistemas de Información

2022 Modalidad Híbrida

Facultad Regional Concepción del Uruguay

Universidad Tecnológica Nacional, Facultad Regional Concepción del Uruguay
10mo. Congreso Nacional de Ingeniería Informática y Sistemas de Información /
compilación de Adrián Callejas ... [et al.]. - 1a ed. - Ciudad Autónoma de Buenos Aires
: Universidad Tecnológica Nacional, 2022.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-950-42-0218-9

I. Tecnología Informática. I. Callejas, Adrián, comp. II. Título.

CDD 004.071

ISBN 978-950-42-0218-9



Este obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.



**10 mo. Congreso Nacional de
Ingeniería Informática / Sistemas de Información**

3 y 4 de Noviembre 2022

Facultad Regional Concepción del Uruguay

UAIRubrics: implementación de una Progressive Web App utilizando una arquitectura serverless

Facundo Romeu

facundonicolas.romeu@uai.edu.ar

Juan Manuel Stecklain

juanmanuel.stecklain@alumnos.uai.edu.ar

Nicolás Battaglia

nicolas.battaglia@uai.edu.ar

*Universidad Abierta Interamericana, Ingeniería en sistemas informáticos
Facultad de Tecnología Informática*

Abstract: *La rúbrica es un instrumento de evaluación que permite determinar el nivel de logro en la resolución de problemas relacionando una serie de indicadores con descriptores de niveles de dominio. Permite evaluar y sistematizar el grado de dominio de las competencias y valorar aspectos complejos, imprecisos y subjetivos. En este trabajo implementamos mediante una aplicación WEB el modelo de rúbricas analíticas que abarca todas las actividades, desde la determinación del objeto de estudio hasta la especificación de sus componentes. Se utilizaron herramientas de última generación teniendo en claro las necesidades y posibilidades, y en base a esto se buscó generar un entorno colaborativo, rápido y seguro en el que el usuario (profesores y alumnos, en principio) manipulen información de forma dinámica y optimizada.*

Palabras clave: rúbricas analíticas, ciencias informáticas, implementación Web

Introducción

Este trabajo se hizo con el fin de desarrollar una implementación WEB para el diseño de rúbricas analíticas que abarca todas las actividades, desde la determinación del objeto de estudio hasta la especificación de sus componentes, el fin de este trabajo es

Diseño de rúbrica analíticas

Carlos Neil [1] propone un modelo de rúbricas que incluye un formato de descriptores analíticos que desambigua su uso y un marco metodológico que asiste en su diseño. Puede ser utilizada en todas las

ampliar y facilitar el diseño y uso de esta herramienta.

Hasta el momento no se tenía conocimiento de una plataforma que permita realizar estas actividades. En cuanto aspectos técnicos se refiere, las herramientas utilizadas se seleccionaron minuciosamente teniendo en cuenta la escalabilidad del proyecto y el buen funcionamiento de este, se espera que el sistema cree un entorno colaborativo y dinámico el cual esté abierto y sea de suma utilidad para profesores, alumnos e interesados. En este trabajo se trabaja en torno al diseño de rúbricas analíticas, luego se inicia la descripción técnica de la aplicación iniciando con el diagrama de casos de uso y realizando una descripción del stack tecnológico utilizado en el sistema, arquitectura, frontend, base de datos y servicios utilizado. Luego se muestra la demostración del alto rendimiento de la aplicación mostrando un test de velocidad realizado. Se finaliza con la definición del futuro de este sistema y como se espera que evolucione el mismo

variantes de la evaluación formativa: la hetero, la auto y coevaluación. En este trabajo se describe la implementación de una aplicación Web para darle soporte al marco metodológico. Como características distintivas, se clarificaron los criterios de evaluación y se

efectiva. El comportamiento de React va de la mano con el Firebase (backend) ya que esta librería utiliza un sistema de estados que son actualizados de forma automática al modificar un elemento almacenado, por lo que el uso de estos estados nos da la capacidad de que al modificar un dato en nuestra base de datos todos los usuarios (que tienen acceso al mismo) vean el impacto automáticamente y no trabajen con datos desactualizados. React también brinda la posibilidad de desarrollar PWA, los beneficios de estas se detallarán más adelante. Al ser una librería con tantas capacidades y beneficios, los usuarios y empresas desarrollan permanentemente herramientas probadas para facilitar la implementación de nuevas funcionalidades, y esto nos brinda un abanico de posibilidades para actualizar a nuevas versiones más rápidamente.

Firebase es una plataforma de Google que brinda herramientas para el desarrollo de aplicaciones. De todas las características que posee, nuestra aplicación utiliza una que es fundamental: la base de datos en tiempo real (Real Time). Elegimos esta base de datos no solo para brindar persistencia, sino que también con el fin de brindar la velocidad de respuesta y actualización de la información buscada, ya que los tiempos de respuesta de esta son extremadamente cortos. Firebase, además, brinda posibilidades como por ejemplo enviar notificaciones PUSH las cuales en un futuro nos servirá para la implementación de nuevas características dentro del sistema.

Arquitectura y Backend

UAI Rubrics está diseñado en arquitectura basada en el modelo de sistemas cliente-servidor. En este caso, el cliente suele denominarse “frontend” y el servidor “backend”. Generalmente, el término backend se refiere a la parte del software que se ejecuta del lado del servidor, lo que también se refiere al trabajo de la parte interna de la aplicación. Esta parte no

interactúa directamente con el usuario, pero es responsable de manipular y guardar los datos y aplicar las reglas de negocio. A continuación, se incluye una explicación de las tecnologías que fueron usadas en el desarrollo de esta aplicación, del lado del backend.

Firestore Realtime Database

La BBDD (base de datos) en tiempo real de Firebase es una BBDD basada en la nube, no relacional que sincroniza y guarda la información entre usuarios en tiempo real. La información permanece disponible aun cuando la aplicación está offline. Es muy rápida y simple de usar.

Las bases de datos en la nube nos permiten construir aplicaciones añadiendo el acceso a la BBDD directamente desde el lado del cliente. Los datos persisten localmente, incluso offline, los eventos en tiempo real continúan ejecutándose, dándole al usuario una experiencia receptiva y, cuando vuelve la conexión, la BBDD en tiempo real sincroniza los cambios locales en los datos con los cambios remotos que ocurrieron mientras el cliente estaba offline. La BBDD en tiempo real permite definir reglas de seguridad, para determinar quiénes pueden escribir o leer la información.

Además, al ser una base de datos NoSQL, posee diferentes optimizaciones comparadas con una BBDD Relacional. La API de BBDD en tiempo real está diseñada para sólo permitir operaciones que pueden ser ejecutadas rápidamente [2]. Como es mencionado en [3] y como fuimos describiendo en los anteriores párrafos, podemos extraer los estos beneficios claves de utilizar la BBDD en tiempo real de Firebase:

- No solo funciona online, sino también cuando los usuarios están offline, almacenando sus cambios y sincronizándose en el momento en que el usuario se conecte nuevamente.

- Se integra bien con la autenticación de Firebase (descrita más adelante) que ayuda a construir un simple e intuitivo modelo para desarrolladores.
- No necesitas usar servidores para compilar las aplicaciones porque la BBDD de Firebase lo permite con móviles y kits de desarrollo de software (SDK) web.
- La sincronización de datos en tiempo real es posible entre cualquier dispositivo.

Otro servicio provisto por Firebase es el de Autenticación. La autenticación de Firebase nos provee SDK y librerías de interfaz de usuario (UI) para autenticar usuarios en la aplicación.

Es posible autenticar usuarios utilizando tanto FirebaseUI como los manualmente integrando uno o varios SDK para añadir métodos de autenticación en la aplicación [2]

En [3] se menciona que la autenticación siempre ha sido crucial desde que se desarrolla en C, C++ y C#. Las nuevas tecnologías no sólo han simplificado la implementación de autenticación, sino que lo hicieron realmente rápido.

Entre los beneficios de utilizar Firebase Authentication se encuentran los siguientes [4]:

- Escribir no más de 10 líneas de código para compilar la aplicación de autenticación usando Firebase
- La autenticación de Firebase permite iniciar sesión usando email y contraseña, autenticación por teléfono, Google, Facebook, Twitter, Github login y muchos más.
- Firebase UI ayuda a customizar la interfaz de usuario de los usuarios al momento de iniciar o cerrar sesión.
- La autenticación de Firebase es muy segura y ha sido desarrollada por el equipo de Google sign-in.

La arquitectura utilizada es una arquitectura serverless que se comunica y almacena los datos en formato JSON, esto facilita y agiliza la comunicación cliente servidor y estandariza la comunicación de datos para interoperar con otros sistemas, además Firebase proporciona herramientas de notificaciones y autenticación que agilizan el proceso de validación de usuarios y la seguridad de estos [5].

Frontend

Para esta aplicación se decidió realizar una Aplicación Web Progresiva (PWA), una tecnología novedosa la cual combina cualidades de las webs tradicionales y las aplicaciones nativas.

Se buscó desarrollar una aplicación flexible en el aspecto de que la misma busque adaptarse al usuario, por esto mismo se decidió desarrollar una PWA. Las características de este tipo de aplicaciones se desarrollarán con más profundidad en el siguiente punto, una de las posibilidades que brindan las PWA es la capacidad de funcionar sin conexión. La PWA fue desarrollada con React ya que esta librería facilita un entorno de desarrollo para las mismas.

PWA

Una Progressive Web App es una aplicación web tradicional con particularidades de apps nativas. Son progresivas ya que buscan aprovechar los requisitos que brinda el entorno en lugar de tener requisitos rígidos [6]. Se puede descargar desde el navegador con solo entrar a la página y tocar un botón. También puede ejecutarse desde el navegador. Se ejecuta como una aplicación, en una ventana de aplicación sin una barra de direcciones u otra interfaz de usuario del navegador. Y como todas las demás aplicaciones instaladas, es una aplicación de nivel superior en el gestor de tareas [7].

- Muy livianas y rápidas.

- Las actualizaciones se instalan de forma instantánea.
- Con solo acceder una sola vez el usuario ya tiene la posibilidad de utilizar la aplicación de forma offline, utilizando su memoria caché.
- La aplicación tiene la posibilidad de aparecer en búsquedas web.
- Existe la posibilidad de utilizar notificaciones PUSH.

Los requerimientos no funcionales las PWA [6] son necesidades de nuestro sistema las cuales no son funcionalidades o herramientas que le permitan realizar acciones al usuario.

En base al contexto y el tipo de aplicación que se está desarrollando se identificaron los siguientes requerimientos no funcionales:

- Garantizar generalmente la accesibilidad, rapidez, y estabilidad de la PWA para lograr un uso eficiente de la memoria RAM.
- Diseñar una interfaz amigable y predecible para una buena experiencia de los usuarios.
- Protección de los datos tanto de usuarios como de la organización.
- El sistema debe ser escalable para futuras implementaciones.

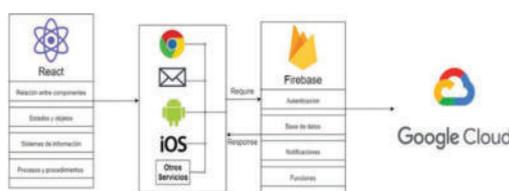


Figura 4. Arquitectura Tecnológica PWA [7]

El cliente de la aplicación fue diseñado en el lenguaje de programación ReactJs, basado en JavaScript, este lenguaje es de los más populares y con más recursos a nivel mundial

JavaScript es un lenguaje de programación con las siguientes características que lo diferencian de los demás [8]:

- Interpretado: Las instrucciones se traducen a código máquina en secuencia en tiempo de ejecución.
- Débilmente tipado: Un lenguaje débilmente tipado no tiene restricciones al momento de almacenar un tipo de dato en una variable, no es necesario la especificación del tipo de dato.
- Orientado a objetos: Un lenguaje de programación orientado a objetos entiende las entidades del programa como clases que pueden inicializarse en objetos.
- Basado en prototipos: los objetos que se utilizan no son instanciados por medio de clases, sino que mediante líneas de código o clonación de objetos
- Dinámico: se busca reducir el tiempo de ejecución de procesos dividiéndolos en subestructuras o subprocessos.
- Imperativo: El código determina línea por línea las instrucciones a llevar a cabo, utiliza estructuras de control para modificar valores y generar resultados deseados.

JavaScript está alineado con nuestro documento HTML y por lo tanto es parte del mismo programa que va a ser utilizado por el cliente [9]. Además, se puede interpretar del lado del servidor utilizando un entorno en tiempo de ejecución multiplataforma como lo es Node.js, basado en el motor V8 de Google, el cual tiene una arquitectura orientada a eventos. Por otro lado, React es una biblioteca open source diseñada para crear aplicaciones single-page (SPA), esta herramienta fue utilizada para diseñar aplicaciones como Instagram. Uno de los puntos fundamentales de React es la posibilidad de desarrollar grandes aplicaciones sin necesidad de recargar la página, es una herramienta rápida y simple de implementar. Por otro lado, utilizamos JSX. Es una extensión para el lenguaje JavaScript usada por React, JSX es implementado en preprocesadores como

babel con el fin de darle al usuario un lenguaje de programación similar a HTML dentro de documentos JavaScript. No es obligatorio usar JSX dentro de React, pero es una herramienta esencial para poder crear componentes.

Para manipular adecuadamente los estados de la aplicación utilizamos Virtual Dom (VDOM). El VDOM es una representación en memoria del DOM real el cual trabaja en sincronía con el mismo, para generar esta copia React utiliza la librería ReactDOM. El DOM es una estructura almacenada en memoria que representa cada uno de los elementos de la página, el dom permite interactuar con cada uno de estos objetos por medio de JavaScript, podemos cambiar tamaños, colores, valores y alterarlos a nuestro gusto. El VDOM herramienta le da la capacidad de modificar estados en la UI de manera inmediata llevando el dom al estado que nosotros necesitemos.

Pruebas de rendimiento:



Figura 5. Resultados del test de performance y tiempo de carga

Esta prueba arrojó resultados positivos, tal como indica la imagen nuestra plataforma se encuentra en el grado A de performance, el cual es el más alto en la escala de la prueba realizada.

A continuación, se detallan los puntos evaluados, los resultados esperados y los resultados obtenidos.

Performance: Tal como se informa en [10] la performance indica el rendimiento de la página desde la perspectiva del usuario, la misma es evaluada en las siguientes métricas.

Rendimiento de carga (45%)

- Primera pintura con contenido (10%): esta es una métrica de rendimiento que mide qué tan rápido los visitantes pueden ver el contenido real (es decir, texto,

imágenes, video, etc.) en su página [10]. En el caso de nuestra plataforma el tiempo en generar esta pantalla es de 1.3 segundos. Índice de velocidad (10%): el índice de velocidad es el tiempo promedio en el que se muestran las partes visibles de la página. El tiempo de respuesta de UAIRubrics, es un tiempo óptimo para la experiencia del usuario.

- Pintura con contenido más grande: La pintura con contenido más grande es un índice que mide el tiempo que tarda en cargar el elemento de contenido de mayor tamaño en hacerse visible para el usuario. El tiempo medido en este índice es óptimo para la experiencia de usuario.

Interactividad (40%)

- Tiempo para ser interactivo, TTI (10%): TTI mide el tiempo más temprano cuando la página está lista de manera confiable para la interactividad del usuario. El tiempo medido en este índice es de 1.3 segundos, este resultado es el esperado para que la experiencia del usuario sea buena.
- Tiempo total de bloqueo (30%): esta métrica mide la cantidad total de tiempo que la página web estuvo bloqueada, impidiendo que el usuario interactúe con su página. En nuestro caso el tiempo total de bloqueo es nulo y es un punto sumamente importante en la experiencia del usuario que buscamos no aumentar.

Estabilidad Visual (15%)

- Cambio de diseño acumulativo (15%): El cambio de diseño acumulativo mide el cambio inesperado de elementos web mientras se procesa la página. En nuestro caso el valor obtenido está dentro del estándar para una buena experiencia de usuario

- Estructura: Este índice informa qué tan bien está construida la página para un rendimiento óptimo. En el caso de nuestra plataforma la estructura es de un 87% y está dentro del estándar óptimo para la buena experiencia del usuario. En futuras versiones se mejorará la estructura con la incorporación de la compresión de archivos de texto y la eliminación de recursos que pueden bloquear el renderizado, con estas mejoras se estima que la estructura superará el 90%.
 - Web vitales: son métricas formadas por la pintura con contenido más grande, el tiempo total de bloqueo y el cambio de diseño acumulativo. Estas métricas centrales indican si está brindando una experiencia rápida y agradable a sus visitantes. Actualmente los resultados obtenidos están dentro de los parámetros que Google considera óptimos [11].
- Escalabilidad: Se espera que el sistema crezca el volumen de usuarios y en paralelo también se contempla el lanzar al público versiones con más funcionalidades.
 - Interacción: El sistema busca ser lo más interactivo y dinámico posible aprovechando las funcionalidades que nos brinda una PWA desarrollada con Firebase.

Estos puntos fueron las bases del sistema desarrollado y en base a los mismos se seleccionaron las tecnologías a utilizar. La plataforma está en camino a ser un entorno dinámico en el que profesores y alumnos puedan no solo trabajar con rúbricas, sino que también en futuras versiones poder entablar un entorno colaborativo en el que compartir y versionar las mismas. Para lograr estos resultados, esperamos utilizar el marco de trabajo de Firebase junto a otras herramientas que permitirán mantener sincronizados todos los usuarios con información en tiempo real

CONCLUSIONES Y TRABAJOS FUTUROS

En conclusión, se buscó desarrollar una plataforma teniendo en cuenta el modelo de rúbricas analíticas y se encontraron ciertos aspectos técnicos que son fundamentales y de relevancia para una aplicación de estas características:

- Flexibilidad: Se busca que el sistema se adapte lo más posible al contexto que presenta el usuario.
- Optimización: El sistema tiene que ser lo más rápido y estar lo más optimizado posible para reducir los requerimientos del usuario y ser capaz de manejar grandes cantidades de datos en simultáneo.
- Experiencia de usuario: la plataforma tiene una interfaz simple e intuitiva.
- Seguridad: Los datos del usuario e información almacenada es segura

Agradecimientos

Ambos queremos agradecer al profesor Prinzo Mauricio Humberto de la materia, Lenguajes de última generación por inculcarnos buenas prácticas y metodologías de trabajo, a nuestras familias por acompañarnos en nuestro proceso de aprendizaje y también agradecer especialmente a Carlos Gerardo Neil y Nicolas Battaglia por ser nuestros acompañantes, tutores y brindarnos una posibilidad tan linda como la de realizar este trabajo con el respaldo de la institución.

Referencias

[1] Neil, C., Battaglia, N., De Vincenzi, M. Edutec. Revista Electrónica. 2022, "Marco metodológico para el diseño de rúbricas analíticas," Edutec Revista Electrónica edutec.es,

- vol. 80, pp. 198–215, 2022, doi: 10.21556/edutec.2022.80.2425.
- [2] Firebase Documentation. “Firebase”.
- [3] Why Firebase is the Best as a Mobile App Development Backend?. “Trista Technology”.
- [4] Firebase Documentation. “Firebase Authentication”.
- [5] Tanna, M., & Singh, H. (2018). *Serverless Web Applications with React and Firebase: Develop real-time applications for web and mobile platforms*. Packt Publishing Ltd.
- [6] Ibañez Pineda, F. J. (2019). *Desarrollo de una Progressive Web App (PWA) para gestionar plataformas tecnológicas modeladas con BPM* (Doctoral dissertation, Universidad Católica de Pereira).
- [7] Google. “Google Codelabs”. [Online] Retrieved September 3, 2022, from <https://developers.google.com/web/fundamentals/codelabs/your-first-pwapp>.
- [8] Maza, M. Á. (2001). JavaScript. En M. Á. Maza, JavaScript (págs. 9-10). España: Innovación y cualificación.
- [9] Zapata Loria, P. M., Rodríguez Huerta, J. F., Ceballos Gómez, S. G., & Mis May, J. R. Aplicación móvil para el cálculo de posición de mecanismos planos de cuatro barras. in *XVIV congreso nacional de ingeniería eléctrica electrónica del MAYAB*. noviembre del 2019 ISSN 1665-0271 (p. 60).
- [10] Everything you need to know about the new GTmetrix Report (powered by Lighthouse). (2020). GTmetrix.
- [11] Walton, P. (2020, July 21). Web Vitals.