



Actividad

Sprint 5. GraphQL y NodeJs y aplicación de interfaz gráfica con ReactJs

Desarrollo de aplicaciones web

1. Título:

Sprint 5. GraphQL, NodeJs y ReactJs y aplicación de la interfaz gráfica con ReactJs

2. Modalidad:

Célula de desarrollo (trabajo en grupo de cinco integrantes)

3. Resultado de aprendizaje:

Se espera que, al finalizar la actividad, los estudiantes sean capaces de crear Schemas para actualizar y devolver la información albergada en la base de datos previamente almacenada en los sprint anteriores sobre MongoDB. Además, deberán ser capaces de poner en práctica todo el conocimiento, creando un servidor NodeJs con el framework Express para ingresar, actualizar y consultar información mediante los diferentes Schemas que se requieren crear para interactuar con la información de la base de datos.

El estudiante tendrá la posibilidad de crear formas para las interfaces gráficas de los CRUD de los proyectos y los integrantes, así como también los diferentes docentes que se requieren para administrar la información.

4. Recursos:

Para lograr la creación de un Schemas, deben ingresar a los enlaces que se presentan y acceder a los recursos indicados.

Enlace para bajar el instalador NodeJs del sitio:

<https://nodejs.org/en/>



Deben seguir las instrucciones en el documento del curso: GraphQL con NodeJs.pptx consignados en la semana 7.

También debe acceder a los recursos que se encuentran en la semana 6, como son los materiales de estudio y el material de apoyo para el desarrollo de la interfaz gráfica con ReactJs

Indicaciones:

- Deben continuar con el mismo grupo de trabajo creado en el sprint anterior (cinco estudiantes).
- Pueden continuar con los mismos roles designados en el sprint 1, o, si lo prefieren, pueden distribuir nuevamente los roles. Estos roles están definidos en la siguiente tabla:

Rol	Función del rol
1. Desarrollador	Generar las fuentes para el frontend y el backend
2. Analista	Levantar las necesidades del negocio para convertirlas en requerimientos del sistema
3. Product Owner	Conocer el producto <i>software</i> que se va a elaborar en el proyecto
4. Administrador de BD	Gestionar y disponer el motor de base de datos para el proyecto de desarrollo de <i>software</i>
5. Scrum master	Gestionar todas las ceremonias y actividades de la metodología Scrum y asignar los recursos a cada rol

- Cada estudiante debe desarrollar las actividades correspondientes para cumplir con las historias de usuario, como se muestra a continuación. Se deben aplicar todas las historias de usuarios que se presentan en este documento, siguiendo las instrucciones en el apartado de **recursos**.

Cada Feature va acompañado con sus diferentes historias de usuarios (HU).



Feature: como analista/administrador **necesito** definir el **Schemas** de GraphQL **para** tener la consulta de los proyectos almacenados en la base de datos MongoDB.

Historia de usuario: HU_022

Dado que tengo información almacenada en la base de datos de los proyectos

Cuando consulte con el Schema definido de proyectos

Entonces veré toda la información de los proyectos almacenados en la base de datos

Historia de usuario: HU_023

Dado que tengo información almacenada en la base de datos de los proyectos

Cuando consulte con el Schema definido de proyectos para un proyecto en particular

Entonces veré todos los campos del proyecto almacenado en la base de datos

Feature: como analista/administrador **necesito** definir el **Schemas** de GraphQL **para** tener la información de los participantes de los proyectos almacenados en la base de datos MongoDB

Historia de usuario: HU_024

Dado que tengo información almacenada en la base de datos de los integrantes de los proyectos de investigación

Cuando consulte con el Schema definido de integrantes

Entonces veré toda la información de los integrantes que participan en los proyectos de investigación

Historia de usuario: HU_025

Dado que tengo información almacenada en la base de datos de los integrantes de los proyectos de investigación



Cuando consulte con el Schema definido de integrantes para un integrante en particular

Entonces verá todos los campos del integrante que participan en el proyecto de investigación

Feature: como analista/administrador **necesito** definir los **Schemas** de GraphQL **para** *gestionar* la información de los nuevos proyectos

Historia de usuario: HU_026

Dado que tengo información de los proyectos para ser creada en la base de datos

Cuando cree un proyecto con el Schema de mutación

Entonces se almacenará la información del proyecto en base de datos

Historia de usuario: HU_027

Dado que tengo información del proyecto para ser actualizada

Cuando actualice un proyecto con el Schema de mutación

Entonces se actualizará la información del proyecto

Historia de usuario: HU_028

Dado que tengo información del proyecto para eliminar

Cuando elimine un proyecto con el Schema de mutación

Entonces se borrará la información del proyecto

Feature: como analista/administrador **necesito** definir los **Schemas** de GraphQL **para** *gestionar* la información de los *integrantes* de los proyectos

Historia de usuario: HU_029

Dado que tengo información de los integrantes para ser creada

Cuando cree un integrante con el Schema de mutación

Entonces se almacenará la información del integrante en base de datos



Historia de usuario: HU_030

Dado que tengo información del integrante para ser actualizada
Cuando actualice un integrante con el Schema de mutación
Entonces se actualizará la información del integrante

Historia de usuario: HU_031

Dado que tengo información del integrante para eliminar
Cuando elimine un integrante con el Schema de mutación
Entonces se borrará la información del integrante

Feature: como analista/administrador **necesito** definir los Schemas de GraphQL **para** *adicionar integrantes* a los proyectos de investigación

Historia de usuario: HU_032

Dado que tengo información del integrante para agregar a un proyecto
Cuando agregue un integrante a un proyecto con el Schema de mutación
Entonces se actualizará la información del proyecto con los integrantes

Historias de usuario para la interfaz gráfica

Feature: como analista **necesito** definir las interfaces gráficas **para** gestionar el CRUD de los proyectos mediante ReactJs

Historia de usuario: HU_033

Dado que tengo la información de los proyectos
Cuando requiera gestionar un proyecto
Entonces podré utilizar la interfaz gráfica para el manejo de la información de los proyectos

Feature: como analista **necesito** definir las interfaces gráficas **para** gestionar el CRUD de los integrantes mediante ReactJs



Historia de usuario: HU_034

Dado que tengo la información de los integrantes de los proyectos
Cuando requiera gestionar los datos de los integrantes
Entonces podré utilizar la interfaz gráfica para el manejo de la información de los integrantes

Feature: como analista **necesito** definir las interfaces gráficas **para** gestionar los maestros necesarios del proyecto mediante Reactjs

Historia de usuario: HU_035

Dado que tengo las pantallas de los maestros
Cuando requiera gestionar la información de los diferentes maestros
Entonces podré utilizar la interfaz gráfica para el manejo de la información de todos los maestros

En Scrum se deben realizar entregas parciales y regulares de un producto final, y las actividades se deben priorizar de acuerdo a las necesidades para lograr la entrega de un producto completo.

En cada uno de los cinco *sprints* se debe desarrollar la temática planteada. De igual manera, el equipo debe entregar el *sprint* en la semana asignada. Esta entrega recibirá una nota que será el soporte para la certificación, pero, aunque el trabajo sea grupal, la nota será de manera individual.

5. Criterios de valoración de la evidencia:

Deben entregar evidencias de un servidor funcional de Nodejs con Express para resolver toda la consulta solicitada en las historias de usuario (HU) y las interfaces gráficas mínimas en reactjs para gestionar la información de los proyectos. Esta actividad tiene un valor del 20% y debe ser entregada en la semana 7.



6. Protocolo de entrega

- **Formato:** documento con:
 - Portada con los nombres de los integrantes del grupo.
 - Documento evidenciando las pantallas en React.
 - Descripción del proceso que se llevó a cabo para cumplir con los requerimientos de este sprint.
 - Evidencia de la funcionalidad de aplicación interfaz gráfica
 - Aportes de cada uno de los integrantes del proyecto.
 - El archivo debe nombrarse como Sprint5_Equipo_XX, en el que XX depende del nombre que le asignaron al equipo grupo.
- **Medio:**

Recurso **tarea**, disponible en el aula virtual, semana 7, en la plataforma Moodle.
- **Plazo máximo de entrega:**

De acuerdo con el cronograma de actividades (entrega semana 7).