

## **MANUAL TÉCNICO DEL SISTEMA**

## CONTENIDO

	Pág.
PRESENTACIÓN.....	5
RESUMEN.....	6
OBJETIVO.....	7
FINALIDAD DEL MANUAL .....	7
INTRODUCCIÓN.....	8
1. ASPECTOS TÉCNICOS.....	9
1.1. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO .....	9
1.1.1. Visual Studio Code.....	9
1.1.2. GitHub.....	9
1.1.3. Node.js.....	10
2. DIAGRAMAS DE MODELAMIENTO .....	11
2.1. DIAGRAMA ENTIDAD RELACION.....	11
2.2. DIAGRAMA LOGICO.....	12
2.3. DICCIONARIO DE DATOS.....	13
3. ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA.....	14
3.1. MODIFICACIÓN LOCAL.....	14
4. REQUERIMIENTOS DEL SOFTWARE.....	21
4.1. REQUISITOS MÍNIMOS.....	21
BIBLIOGRAFÍA.....	22

## LISTA DE FIGURAS

	Pág.
Figura 1. Diagrama de modelado de clases.....	11
Figura 2. Diagrama de casos de uso .....	12
Figura 3. Página web de descarga Node.js.....	14
Figura 4. Página web de descarga Visual Studio Code .....	15
Figura 5. Página web de descarga GitHub.....	15
Figura 8. Validación del archivo package.json .....	17
Figura 9. Ejecución de comando para instalar paquetes necesarios .....	17

**LISTA DE TABLAS**

	Pág.
Tabla 1. Diccionario de datos.....	13

## **PRESENTACIÓN**

El siguiente manual se ha desarrollado con la finalidad de dar a conocer la información necesaria para realizar mantenimiento, instalación y exploración del software el cual consta de diferentes actividades para el mejoramiento de los procesos de psicomotricidad gruesa y concentración en los niños con síndrome de Down del Colegio Espíritu Santo Marianista.

El manual ofrece la información necesaria de ¿cómo está realizado el software? para que la persona (Desarrollador en el framework DJANGO) que quiera editar el software lo haga de una manera apropiada, dando a conocer la estructura del desarrollo del aplicativo.

## **RESUMEN**

El manual detalla los aspectos técnicos e informáticos del software con la finalidad de explicar la estructura del aplicativo al personal que quiera administrarlo, editarlo o configurarlo. La siguiente guía se encuentra dividida en las herramientas que se usaron para la creación del software con una breve explicación paso a paso, El aplicativo web maneja diferentes funcionalidades el cual requieren de hardware y software el cual se explicará que funcionamiento realiza cada uno de ellos, dando sugerencias para el debido uso del sistema de información.

Manual técnico del sistema

## **OBJETIVO**

Dar a conocer el uso adecuado del software en aspectos técnicos de manera descriptiva e ilustrada sobre los componentes y funcionalidades que conforman el buen funcionamiento del sistema de información.

## **FINALIDAD DEL MANUAL**

La finalidad de éste manual técnico es instruir a la persona que quiera administrar, editar o configurar el software usando las debidas herramientas.

## INTRODUCCIÓN

El manual se realiza con el fin de detallar el software en términos técnicos para que la persona que vaya a administrar, editar o configurar el aplicativo lo haga de una manera apropiada. El documento se encuentra dividido en las siguientes secciones:

- **ASPECTOS TEÓRICOS:** Se darán a conocer conceptos, definiciones y explicaciones de los componentes del aplicativo desde un punto de vista teórico para mayor entendimiento por parte del lector sobre el funcionamiento del sistema de información e herramientas.
- **DIAGRAMAS DE MODELAMIENTO:** Se compone por diagramas e ilustraciones alusivos al funcionamiento del aplicativo.
- **ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA:** Corresponde a la instrucción al lector sobre los componentes del aplicativo desde una perspectiva técnica en los aspectos de almacenamiento de datos, estructura del desarrollo y recomendaciones del uso debido del aplicativo.
- **REQUERIMIENTOS DEL SOFTWARE:** Detalla los requerimientos básicos necesarios para el funcionamiento del software.



## **1. ASPECTOS TÉCNICOS**

### **1.1. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO**

La finalidad de una página web de artesanos de una universidad es proporcionar un espacio en línea para que los artesanos puedan mostrar sus productos y habilidades a un público más amplio. La página web también puede ayudar a los artesanos a establecer contactos y oportunidades de venta con compradores interesados en sus productos. Además, una página web bien diseñada puede proporcionar información detallada sobre los productos, los procesos de producción y los artistas, lo que ayuda a los visitantes a comprender mejor el valor y la calidad de los productos artesanales.

Es ésta sección se procede a explicar las herramientas informáticas empleadas para el desarrollo del aplicativo:

#### **1.1.1. Visual Studio code**

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es un editor de código fuente gratuito y de código abierto que admite una amplia variedad de lenguajes de programación y marcos. Se puede ejecutar en Windows, macOS y Linux.

VS Code es altamente personalizable y está diseñado para ser una herramienta fácil de usar y eficiente para desarrolladores de software. Ofrece características útiles como resaltado de sintaxis, completado automático de código, depuración, refactorización, control de versiones y soporte para extensiones. Además, VS Code se integra fácilmente con herramientas populares de desarrollo web como Git, Node.js y TypeScript. Con su enfoque en la productividad y la eficiencia, VS Code se ha convertido en una opción popular entre los desarrolladores de todo el mundo.

#### **1.1.2. GitHub**

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo,

## Manual técnico del sistema

dentro de un proyecto. Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls. Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original. (Luciano Castillo, 2012)

### 1.1.3. MySQL Workbench

MySQL Workbench es una herramienta de diseño visual de bases de datos y de administración de MySQL. Proporciona una interfaz gráfica para crear, modificar y administrar bases de datos MySQL, lo que facilita la tarea de diseñar y mantener bases de datos complejas. Con MySQL Workbench, los usuarios pueden diseñar modelos de datos, crear diagramas ER (Entidad-Relación), administrar usuarios y permisos, realizar consultas SQL y mucho más. También ofrece una variedad de herramientas útiles para la monitorización del rendimiento, la copia de seguridad y la restauración de bases de datos. En resumen, MySQL Workbench es una herramienta integral para los desarrolladores y administradores de bases de datos MySQL.

## 2. DIAGRAMAS DE MODELAMIENTO

### 2.1. DIAGRAMA ENTIDAD RELACION

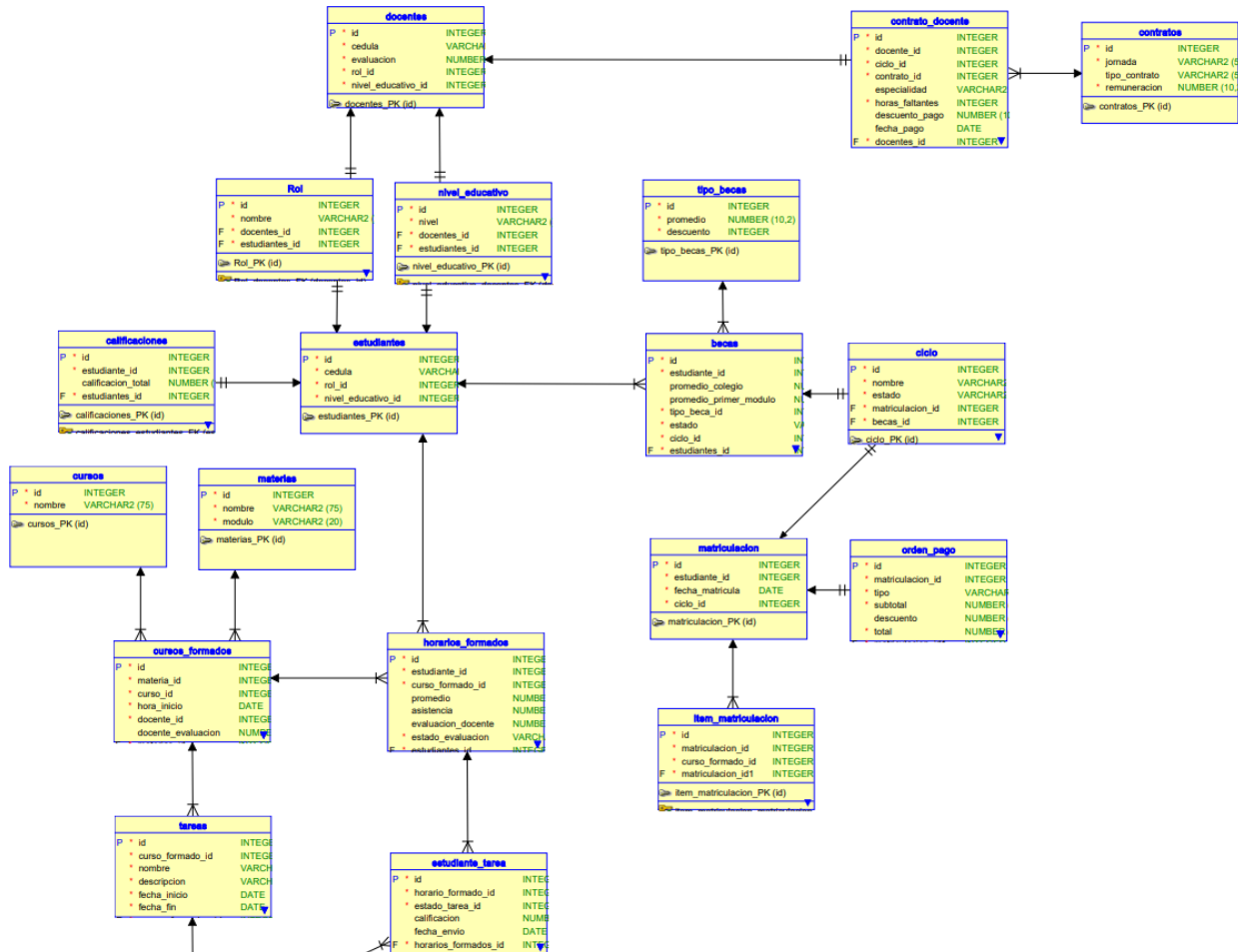


Figura 1. Diagrama de entidad relación

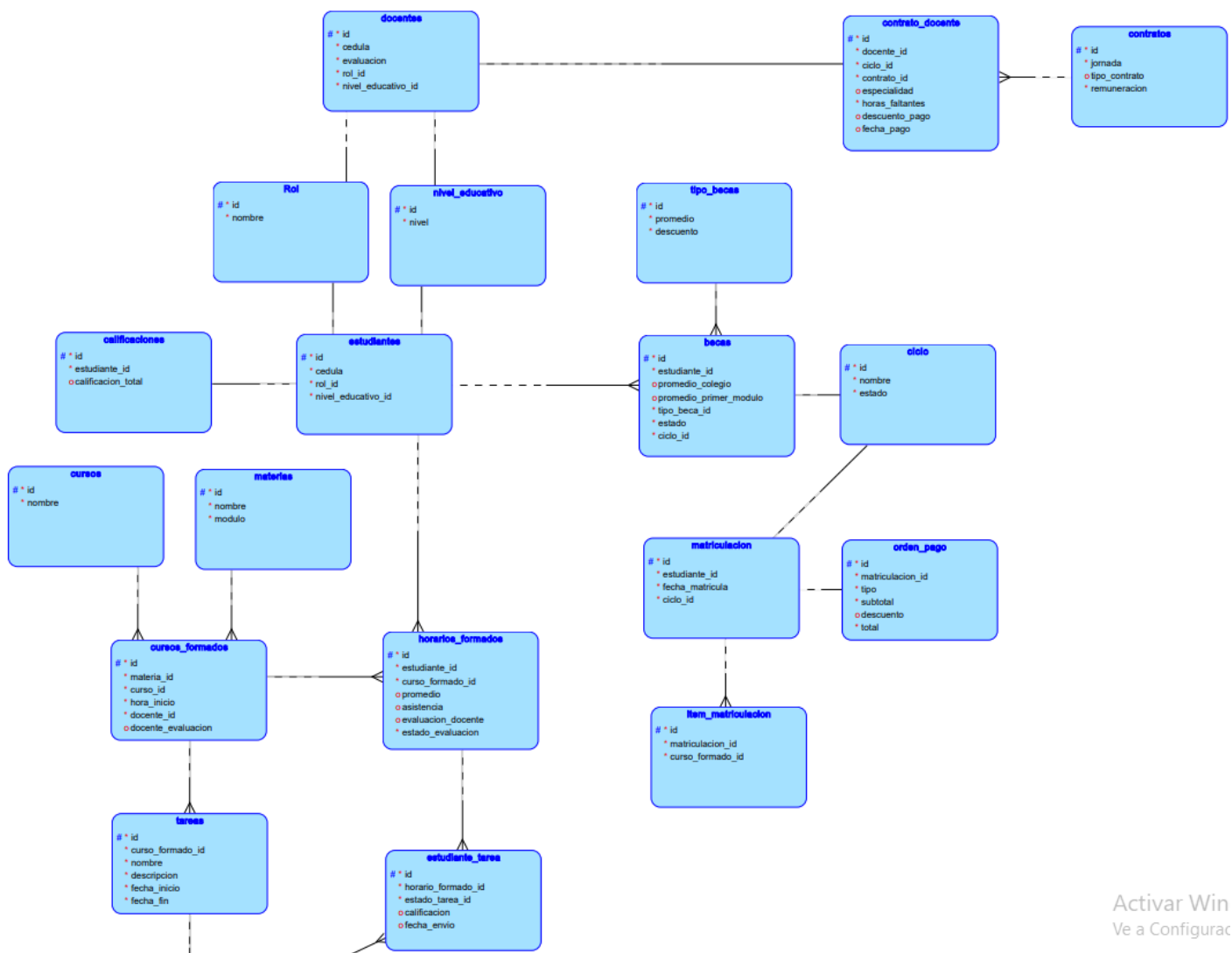
Fuente: Por los autores.

En la figura 1, se detalla cada una de las entidades (modelos) usadas para el almacenamiento de datos del aplicativo, en el cual cada una de ellas realiza las siguientes funciones:

## 2.2. DIAGRAMA RELACIONAL

Un diagrama relacional de base de datos es una representación gráfica que muestra las tablas de una base de datos, así como las relaciones y las restricciones que existen entre ellas.

Figura 2. Diagrama relacional



Fuente: Por los autores.

### 2.3. DICCIONARIO DE DATOS

Para el almacenamiento de datos del software, se definen los campos necesarios para cada una de las entidades relacionadas con el aplicativo.

**Tema: Diccionario de Datos.**

**Tabla 1. Diccionario de datos modelo rol**

rol			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
nombre	VARCHAR	20	cadena de caracteres, no nulo.

**Tabla 2. Diccionario de datos modelo nivel\_educativo**

nivel_educativo			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
nivel	VARCHAR	50	cadena de caracteres, no nulo.

**Tabla 3. Diccionario de datos modelo ciclo**

ciclo			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
nombre	VARCHAR	50	cadena de caracteres, no nulo.
estado	VARCHAR	20	cadena de caracteres, no nulo y valor predeterminado "Inactivo", con restricción de verificación que solo permite "Activo" o "Inactivo".

**Tabla 4. Diccionario de datos modelo contratos**

contratos			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
jornada	VARCHAR	50	cadena de caracteres, no nulo con restricción de verificación que solo permite "Completa" o "Medio tiempo".

## Manual técnico del sistema

tipo_contrato	VARCHAR	50	cadena de caracteres, no nulo con restricción de verificación que solo permite "Nombramiento" o "Contrato".
remuneracion	DECIMAL	10,2	decimal de 10,2, no nulo.

Tabla 5. Diccionario de datos modelo materias

materias			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
nombre	VARCHAR	75	cadena de caracteres, no nulo.
modulo	VARCHAR	20	cadena de caracteres, no nulo, con restricción de verificación que solo permite "Módulo 1" o "Módulo 2".

Tabla 6. Diccionario de datos modelo cursos

cursos			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
nombre	VARCHAR	75	cadena de caracteres, no nulo.

Tabla 7. Diccionario de datos modelo tipo\_becas

tipo_becas			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
promedio	DECIMAL	10,2	decimal de 10,2, no nulo.
descuento	INT		entero, no nulo.

Tabla 8. Diccionario de datos modelo medios

medios			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
medio	VARCHAR	50	cadena de caracteres, no nulo.

Tabla 9. Diccionario de datos modelo tipo\_tarea

tipo_tarea			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
tipo	VARCHAR	50	cadena de caracteres, no nulo.

Tabla 10. Diccionario de datos modelo estudiantes

estudiantes			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		entero, clave principal de la tabla y autonumérico.
cedula	VARCHAR	10	número de cédula del estudiante (cadena de caracteres de longitud 10)
nombres	VARCHAR	100	nombres del estudiante (cadena de caracteres de longitud 100)
apellidos	VARCHAR	100	apellidos del estudiante (cadena de caracteres de longitud 100)
edad	INT		edad del estudiante (entero)
direccion	VARCHAR	150	dirección del estudiante (cadena de caracteres de longitud 150)
email	VARCHAR	150	dirección de correo electrónico del estudiante (cadena de caracteres de longitud 150)
telefono	VARCHAR	10	número de teléfono del estudiante (cadena de caracteres de longitud 10)
contra	VARCHAR	30	contraseña del estudiante (cadena de caracteres de longitud 30)
rol_id	INT		identificador del rol del estudiante (entero)
nivel_educativo_id	INT		identificador del nivel educativo del estudiante (entero)
medio_id	INT		entero, no nulo y con referencia a la tabla medios.

Tabla 11. Diccionario de datos modelo docentes

<b>docentes</b>			
<b>NOMBRE CAMPO</b>	<b>TIPO DATO</b>	<b>TAMAÑO</b>	<b>DESCRIPCIÓN</b>
id	INT		identificador único del docente (entero)
cedula	VARCHAR	10	número de cédula del docente (cadena de caracteres de longitud 10)
nombres	VARCHAR	100	nombres del docente (cadena de caracteres de longitud 100)
apellidos	VARCHAR	100	apellidos del docente (cadena de caracteres de longitud 100)
edad	INT		edad del docente (entero)
direccion	VARCHAR	150	dirección del docente (cadena de caracteres de longitud 150)
email	VARCHAR	150	dirección de correo electrónico del docente (cadena de caracteres de longitud 150)
telefono	VARCHAR	10	número de teléfono del docente (cadena de caracteres de longitud 10)
contra	VARCHAR	30	contraseña del docente (cadena de caracteres de longitud 30)
evaluacion	DECIMAL	10,2	evaluación del docente (número decimal con precisión 10 y escala 2)
rol_id	INT		identificador del rol del docente (entero)
nivel_educativo_id	INT		identificador del nivel educativo del docente (entero)

Tabla 12. Diccionario de datos modelo cursos\_formados

<b>cursos formados</b>			
<b>NOMBRE CAMPO</b>	<b>TIPO DATO</b>	<b>TAMAÑO</b>	<b>DESCRIPCIÓN</b>
id	INT		identificador único del curso formado (entero)
materia_id	INT	50	identificador de la materia del curso formado (entero)
curso_id	INT		identificador del curso del curso formado (entero)
hora_inicio	TIME		hora de inicio del curso formado (tiempo)



## Manual técnico del sistema

cupo	INT		cupo máximo de estudiantes para el curso formado (entero)
docente_id	INT		identificador del docente que forma el curso (entero)
docente_evaluacion	DECIMAL	10,2	evaluación del docente del curso formado (número decimal con precisión 10 y escala 2)

Tabla 13. Diccionario de datos modelo becas

becas			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		identificador único de la beca (entero)
estudiante_id	INT	50	identificador del estudiante que recibió la beca (entero)
promedio_colegio	DECIMAL	10,2	promedio del estudiante en el colegio (número decimal con precisión 10 y escala 2)
promedio_primer_modulo	DECIMAL	10,2	promedio del estudiante en el primer módulo (número decimal con precisión 10 y escala 2)
tipo_beca_id	INT		identificador del tipo de beca (entero)
estado	VARCHAR	20	estado de la beca (cadena de caracteres de longitud 20, con valores "Activo" o "Inactivo")
ciclo_id	INT		identificador del ciclo en el que se otorgó la beca (entero)

Tabla 14. Diccionario de datos modelo matriculacion

matriculacion			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		identificador único de la matriculación (entero)
estudiante_id	INT		identificador del estudiante matriculado (entero)
fecha_matricula	DATE		fecha de matriculación del estudiante (fecha)

## Manual técnico del sistema

ciclo_id	INT		identificador del ciclo de la matriculación (entero)
----------	-----	--	--

Tabla 15. Diccionario de datos modelo item\_matriculacion

item_matriculacion			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		identificador único del item de matriculación (entero)
matriculacion_id	INT		identificador de la matriculación a la que pertenece el item (entero)
curso_formado_id	INT		identificador del curso formado que se matriculó (entero)

Tabla 16. Diccionario de datos modelo horarios\_formados

horarios_formados			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		identificador único de cada registro de horario formado (entero).
estudiante_id	INT		identificador único del estudiante asociado al horario (entero no nulo, se hace referencia a la tabla estudiantes).
curso_formado_id	INT		identificador único del curso formado asociado al horario (entero no nulo, se hace referencia a la tabla cursos_formados).
promedio	DECIMAL	10,2	promedio del estudiante en el curso formado (decimal con 10 dígitos en total y 2 decimales, valor por defecto es 0.00).
asistencia	DECIMAL	10,2	porcentaje de asistencia del estudiante al curso formado (decimal con 10 dígitos en total y 2 decimales, valor por defecto es 0.00).
dias_asistidos	INT		número de días en que el estudiante asistió al curso formado (entero, valor por defecto es 0).

### Manual técnico del sistema

dias_faltas	INT		número de días en que el estudiante faltó al curso formado (entero, valor por defecto es 0).
evaluacion_docente	DECIMAL	10,2	evaluación numérica del docente del curso formado por el estudiante (decimal con 10 dígitos en total y 2 decimales).
estado_evaluacion	VARCHAR	20	estado de la evaluación del docente del curso formado por el estudiante (cadena de caracteres de hasta 20 caracteres, no nulo, valor por defecto es 'No evaluado'). Sólo pueden tomar los valores 'Evaluado' o 'No evaluado'.

**Tabla 17. Diccionario de datos modelo tareas**

tareas			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		identificador único de cada registro de tarea (entero).
curso_formado_id	INT		identificador único del curso formado asociado a la tarea (entero no nulo, se hace referencia a la tabla cursos_formados).
nombre	VARCHAR	100	nombre de la tarea (cadena de caracteres de hasta 100 caracteres, no nulo).
descripcion	VARCHAR	300	descripción de la tarea (cadena de caracteres de hasta 300 caracteres, no nulo).
fecha_inicio	DATETIME		fecha de inicio de la tarea (fecha y hora, no nulo).
fecha_fin	DATETIME		fecha de finalización de la tarea (fecha y hora, no nulo).
tipo_tarea_id	INT	DEFAULT 1	identificador único del tipo de tarea (entero no nulo, se hace referencia a la tabla tipo_tarea). El valor por defecto es 1.

**Tabla 18. Diccionario de datos modelo estudiante\_tarea**

<b>estudiante_tarea</b>			
<b>NOMBRE CAMPO</b>	<b>TIPO DATO</b>	<b>TAMAÑO</b>	<b>DESCRIPCIÓN</b>
id	INT		identificador único de la tarea del estudiante (entero).
estudiante_id	INT		identificador único del estudiante (entero, no nulo, referencia a estudiantes(id)).
estado_tarea	VARCHAR	20	estado actual de la tarea del estudiante (cadena de caracteres de máximo 20 caracteres, no nulo, valor por defecto "No enviado", valor permitido "Enviado", "No enviado", "Atrasado", "Inactivo").
entrega	VARCHAR	1000	archivo o información relacionada con la entrega de la tarea del estudiante (cadena de caracteres de máximo 1000 caracteres).
tarea_id	INT		identificador único de la tarea (entero, no nulo, referencia a tareas(id)).
calificacion	DECIMAL	10,2	calificación obtenida por el estudiante en la tarea (decimal de 10 dígitos, 2 decimales).
fecha_envio	DATETIME		fecha y hora de envío de la tarea por parte del estudiante (fecha y hora).

Tabla 19. Diccionario de datos modelo calificaciones

calificaciones			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		identificador único de la calificación (entero).
estudiante_id	INT		identificador único del estudiante (entero, no nulo, referencia a estudiantes(id)).
calificacion_total	DECIMAL	10,2	calificación total obtenida por el estudiante (decimal de 10 dígitos, 2 decimales).

Tabla 20. Diccionario de datos modelo orden\_pago

orden_pago			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		identificador único de la orden de pago (entero).
matriculacion_id	INT		identificador único de la matriculación del estudiante (entero, no nulo, referencia a matriculacion(id)).
tipo	VARCHAR	50	tipo de materia o módulo para el cual se está realizando el pago (cadena de caracteres de máximo 50 caracteres, no nulo, valor permitido "Materia individual", "Módulo 1", "Módulo 2").
subtotal	DECIMAL	10,2	subtotal de la orden de pago (decimal de 10 dígitos, 2 decimales, no nulo).
descuento	DECIMAL	10,2	descuento aplicado a la orden de pago (decimal de 10 dígitos, 2 decimales).
total	DECIMAL	10,2	total, de la orden de pago (decimal de 10 dígitos, 2 decimales, no nulo).

Tabla 21. Diccionario de datos modelo contrato\_docente

contrato_docente			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		identificador único del contrato del docente (entero).
docente_id	INT		identificador único del docente (entero, no nulo, referencia a docentes(id)).
ciclo_id	INT		identificador único del ciclo lectivo del contrato (entero, no nulo, referencia a ciclo(id)).
contrato_id	INT		identificador único del contrato en sí (entero, no nulo, referencia a contratos(id)).
especialidad	VARCHAR	100	especialidad del docente (cadena de caracteres de máximo 100 caracteres).

Tabla 22. Diccionario de datos modelo pago\_docente

pago_docente			
NOMBRE CAMPO	TIPO DATO	TAMAÑO	DESCRIPCIÓN
id	INT		identificador único del pago del docente (entero).
contrato_docente_id	INT		identificador único del contrato del docente (entero, no nulo, referencia a contrato_docente(id)).
horas_faltantes	INT		cantidad de horas que faltan para completar las horas totales del contrato del docente (entero, no nulo, valor por defecto 0).
descuento_pago	DECIMAL	10,2	descuento aplicado al pago del docente (decimal de 10 dígitos, 2 decimales)
fecha_pago	DATE		fecha de pago del docente (fecha)

Manual técnico del sistema

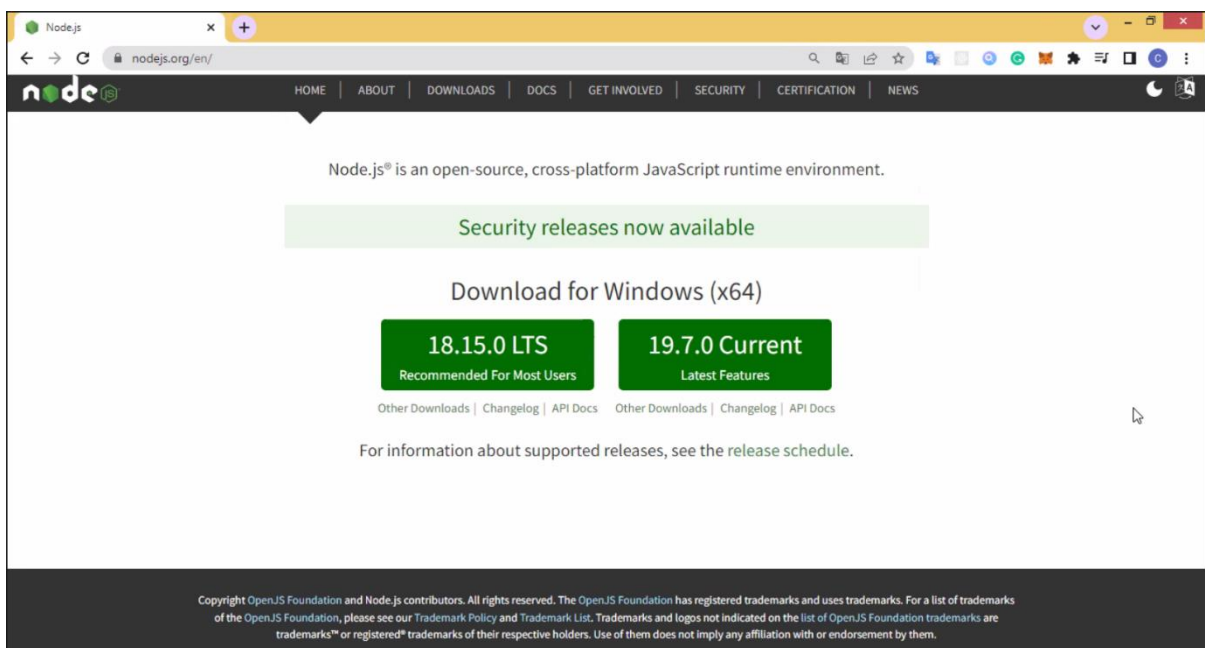
### 3. ASPECTO TÉCNICO DEL DESARROLLO DEL SISTEMA

En la siguiente sección se procede a realizar una descripción detallada sobre los aspectos técnicos del aplicativo, relacionado con la instalación de las herramientas necesarias para realizar modificaciones requeridas de manera ordenada.

#### 3.1. MODIFICACIÓN LOCAL

Si el desarrollador quiere realizar modificaciones del software de manera local, tendrá que realizar la instalación de componentes adicionales, para empezar se debe instalar Node.js 18.15.0 el cual se consigue de manera gratuita en la página [www.nopdejs.org](http://www.nopdejs.org) ,

Figura 3. Página web de descarga Python

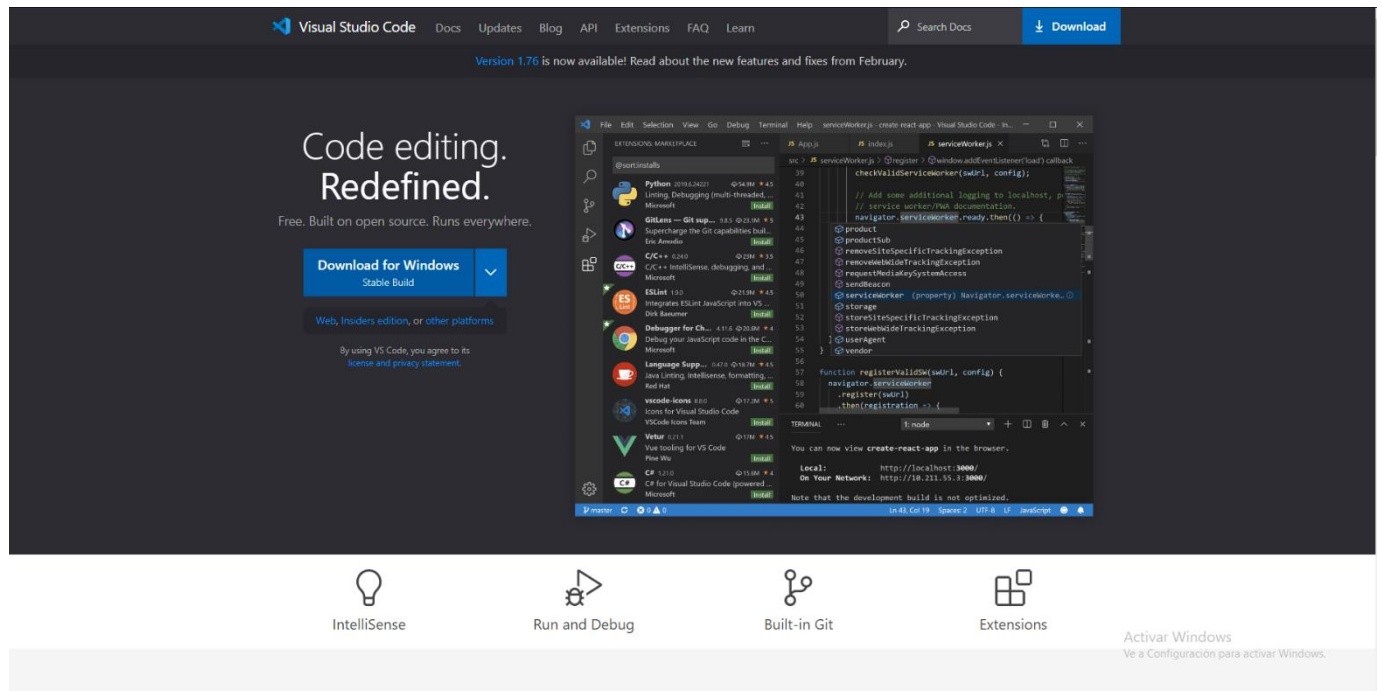


Fuente: (node.js, s.f.)

Al instalar Node.js, procedemos a instalar el editor de texto para código Visual Studio Code, descargamos la última versión en la página oficial, esta descarga es con licencia gratuita.

## Manual técnico del sistema

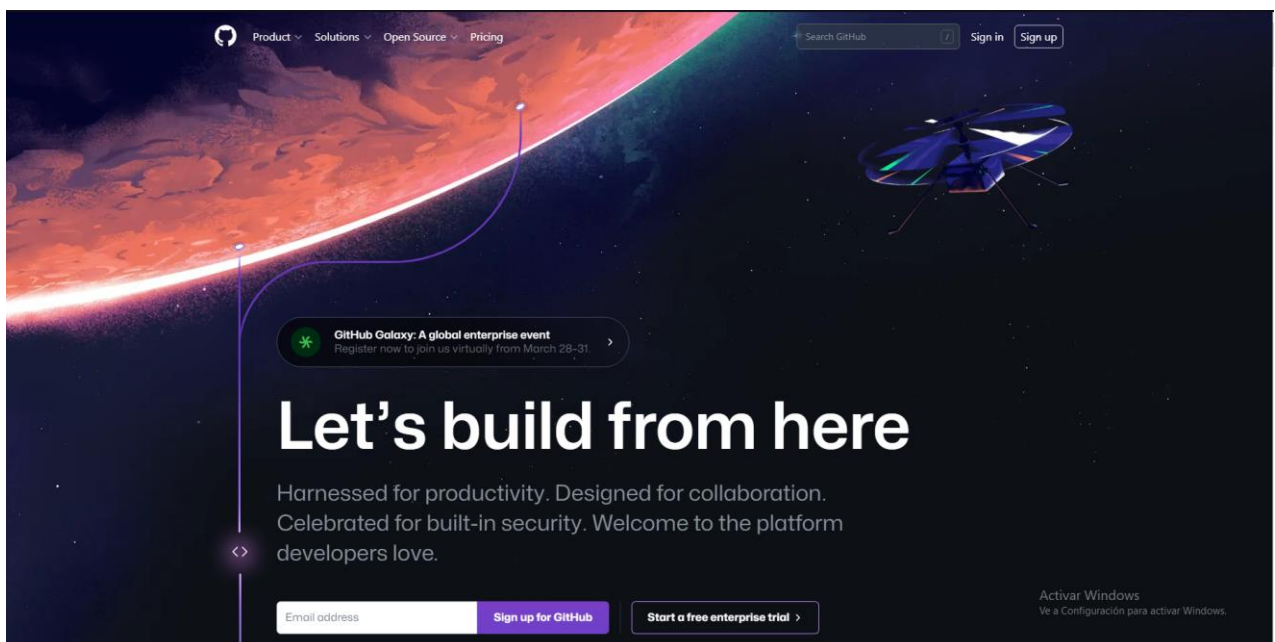
Figura 4. Página web de descarga Visual Studio Code



Fuente: (Visual Studio Code, s.f.)

Se continúa descargando el aplicativo GitHub para escritorio, se realiza desde la página oficial, con el fin de poder obtener el repositorio y realizarle cambios al software llevando un historial.

Figura 5. Página web de descarga GitHub



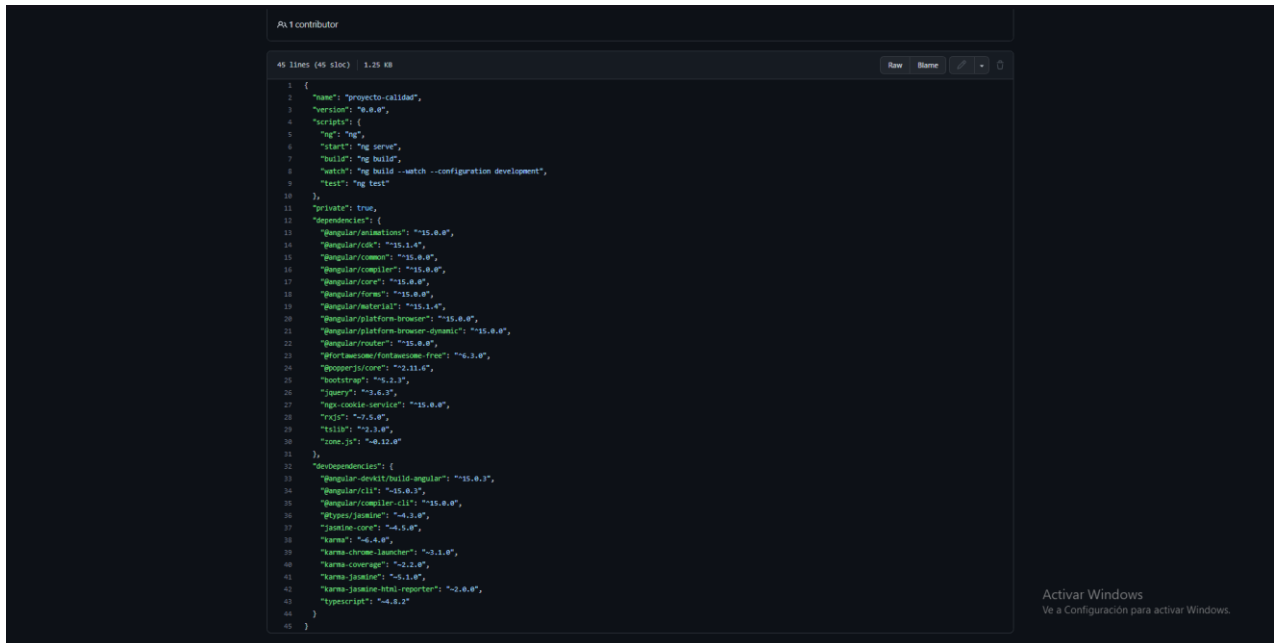
Fuente: (GitHub, s.f.)

en Clone, esa función clona el repositorio en un lugar indicado por el usuario. Se valida que quede en la venta de cambios sin que haya generado error alguno.



## Manual técnico del sistema

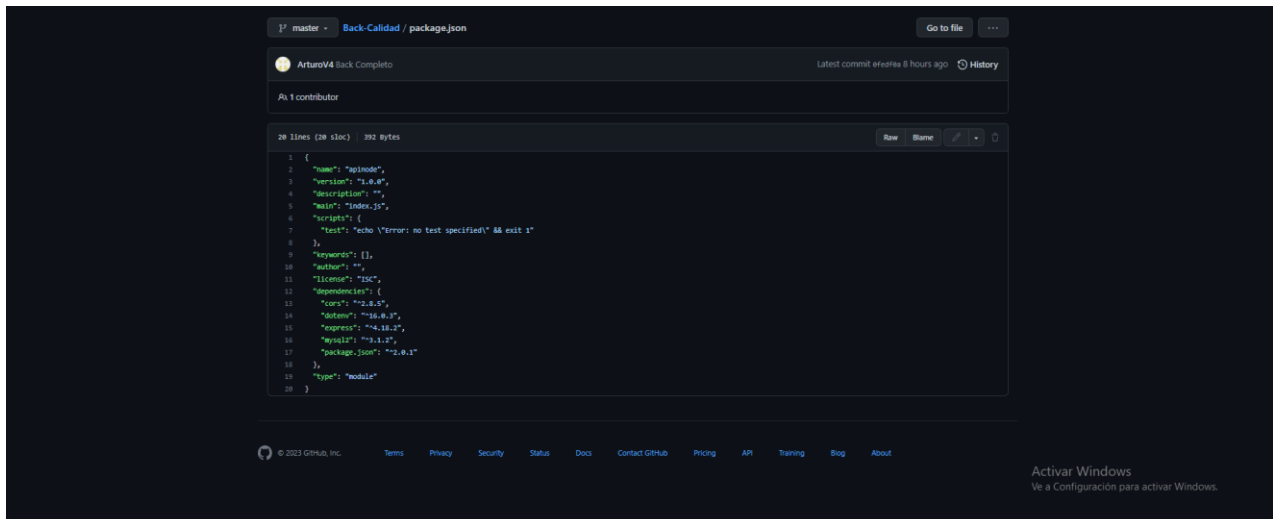
Figura 8. Validación del archivo package.json  
Front-end



The screenshot shows a code editor with a dark theme. The file is named 'package.json' and is 45 lines long. The content is a JSON object representing the project configuration. It includes fields for 'name', 'version', 'scripts', 'private', 'dependencies', and 'devDependencies'. The 'scripts' field contains commands for 'start', 'build', 'watch', and 'test'. The 'dependencies' field lists various Angular and related libraries. The 'devDependencies' field lists development tools like Angular CLI, Karma, and Jasmine.

```
1 {
2   "name": "proyecto-calidad",
3   "version": "1.0.0",
4   "scripts": {
5     "start": "ng serve",
6     "build": "ng build",
7     "watch": "ng build --watch --configuration development",
8     "test": "ng test"
9   },
10  "private": true,
11  "dependencies": {
12    "@angular/animations": "~15.0.0",
13    "@angular/cdk": "~15.1.4",
14    "@angular/common": "~15.0.0",
15    "@angular/compiler": "~15.0.0",
16    "@angular/core": "~15.0.0",
17    "@angular/forms": "~15.0.0",
18    "@angular/material": "~15.1.4",
19    "@angular/platform-browser": "~15.0.0",
20    "@angular/platform-browser-dynamic": "~15.0.0",
21    "@angular/router": "~15.0.0",
22    "@fortawesome/fontawesome-free": "~6.3.0",
23    "@popperjs/core": "~2.11.6",
24    "bootstrap": "~5.2.3",
25    "jquery": "~3.6.0",
26    "ngx-cookie-service": "~15.0.0",
27    "rxjs": "~7.5.0",
28    "tslib": "~2.3.0",
29    "zone.js": "~0.12.0"
30  },
31  "devDependencies": {
32    "@angular-devkit/build-angular": "~15.0.3",
33    "@angular/cli": "~15.0.3",
34    "@angular/compiler-cli": "~15.0.0",
35    "@types/jasmine": "~4.3.0",
36    "jasmine-core": "~4.5.0",
37    "karma": "~6.4.0",
38    "karma-chrome-launcher": "~3.1.0",
39    "karma-coverage": "~2.2.0",
40    "karma-jasmine": "~5.1.0",
41    "karma-jasmine-html-reporter": "~2.0.0",
42    "typescript": "~4.8.2"
43  }
44 }
```

## Back-end



The screenshot shows a code editor with a dark theme. The file is named 'package.json' and is 20 lines long. The content is a JSON object representing the project configuration. It includes fields for 'name', 'version', 'description', 'main', 'scripts', 'dependencies', and 'type'. The 'scripts' field contains a command for 'test'. The 'dependencies' field lists various Node.js and related libraries.

```
1 {
2   "name": "api-node",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC",
12  "dependencies": {
13    "cors": "~2.8.5",
14    "dotenv": "~16.0.3",
15    "express": "~4.18.2",
16    "mysql2": "~3.1.2",
17    "package.json": "~2.0.1"
18  },
19  "type": "module"
20 }
```

Fuente: Por los autores.

Al validar los paquetes, procedemos a abrir el símbolo del sistema (cmd) para ejecutar el comando `npm install package.json` con el fin de instalar los paquetes necesarios para poder inicializar el servidor

## **4. REQUERIMIENTOS DEL SOFTWARE**

En esta sección se detallará los requisitos mínimos del sistema para poder ejecutar los aplicativos usados para modificar el software.

### **4.1. REQUISITOS MÍNIMOS**

- Sistema Operativo: Windows 7
- Procesador: Intel Core Celeron
- Memoria RAM: 1GB
- Disco Duro: 1GB
- Resolución de pantalla: 1280 x 720 pixeles
- Periféricos: Teclado, ratón, Bocinas (Opcional)

## Manual técnico del sistema

### **BIBLIOGRAFÍA**

- Workbench: Oracle Corporation. (2021). MySQL Workbench. Recuperado el 8 de marzo de 2023, de <https://www.mysql.com/products/workbench/>
- Visual Studio Code: Microsoft Corporation. (2021). Visual Studio Code. Recuperado el 8 de marzo de 2023, de <https://code.visualstudio.com/>
- diagrama relacional de base de datos: Elmasri, R., & Navathe, S. B. (2016). Fundamentals of Database Systems (7ª ed.). Pearson

Manual técnico del sistema

Manual técnico del sistema

Manual técnico del sistema