

**Test de Rafael Adrian Herrera Cruz**  
**rafaelhecr@protonmail.com**

**Javascript**

La pregunta 3 note que los cuadros blancos siguen la secuencia de números primos siendo  $m.c.d(n)=1$  o  $n \leftrightarrow n/1$  o  $n/n$   $D \in \{n \Rightarrow 2 \mid R > 3 \text{ Prim.}\}$ , Los cuadros negros siguen la secuencia  $n^2$

La pregunta 4 y 5 siguen la secuencia de Fibonacci  $+ n \Rightarrow f = (f_{n-1} + f_{n+1}) + n$

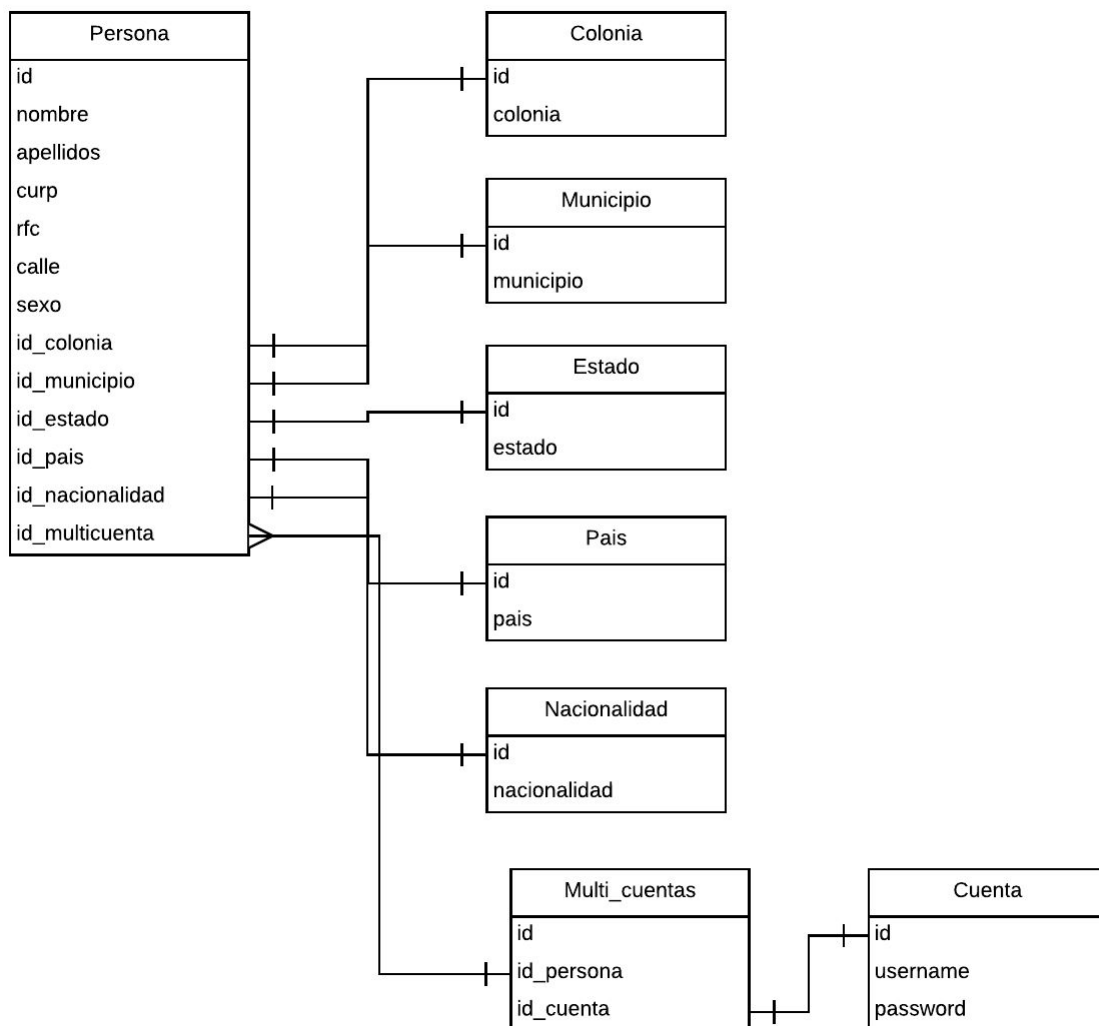
**Python**

Se realizaron las mismas soluciones en Python de JS.

## Bases de datos

1. Las bases de datos SQL como MySQL o MS SQL Server son bases de datos con una estructura rígida, manipulable con un lenguaje y su escalamiento vertical es muy fácil, en cambio las bases de datos NoSQL como MongoDB y Cassandra almacena datos en documentos JSON, fijados por llaves y su manipulación es por métodos javascript, son muy rápidas, flexibles a la estructura y permiten ser escaladas horizontalmente de una forma fácil.

2.



### 3. Personas <Collection>

```
{
  "nombre": "nombre_persona",
  "apellidos": "apellidos_persona",
  "crup": "curp_persona",
  "rfc": "rfc_persona",
```

```

"sexo": "sexo_persona",
"direccion_1": {
  "calle": "calle_domicilio",
  "colonia": "colonia_domicio",
  "municipio": "municipio_domicilio",
  "estado": "estado_domicilio",
  "pais": "pais_domicilio",
  "nacionalidad": "nacionalidad"
},
"cuentas":{
  "cuenta_1":{
    "username": "username",
    "password": "password"
  }
}
}

```

4. DDL es el Lenguaje de Definición de Datos, permite ejecutar instrucciones para crear la estructura de la BD como las tablas y sus relaciones.

5. DML es el Lenguaje de Manipulación de Datos, permite trabajar con los datos que se insertaran en la estructura, con DML se insertan, modifican, eliminan o se consultan.

6. Funciones de agregación: Permiten obtener un valor agregado único de un conjunto de datos, permiten contar, sumar, saber la media o el promedio de una columna, en el caso de contar se pueden agregar varias columnas.

Funciones de agrupación: Permiten agrupar a los datos cumpliendo características que se soliciten en la consulta ordenando la comprensión y permitiendo extraer información útil de un gran conjunto de datos.

7. En respuesta al teorema CAP es cuando la base de datos sigue respondiendo incluso cuando esta está hecha para trabajar bajo backups y la comunicación entre diferentes equipos deja de funcionar y la sincronización entre estas BD fallan, los datos al usuario siguen disponibles.

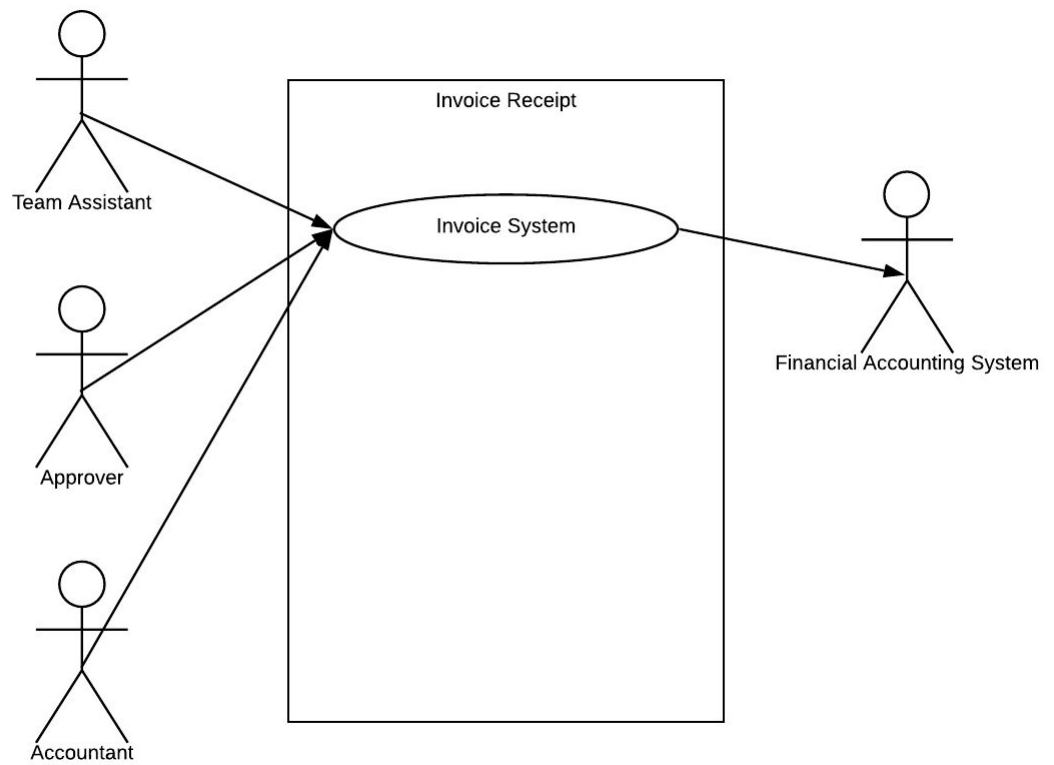
8. De nuevo usando el teorema CAP es cuando se usan diferentes nodos para consultar los datos NoSQL y cuando uno cae, la funcionalidad sigue siendo correcta.

## Testing

1 y 2. Se encuentra la bateria de pruebas en JS con Mocha, dentro de la carpeta Test.

3.

4.



## Preguntas

1. La verificación en informática es asegurarse el el flujo y los datos hagan lo que realmente se esperan que hagan y se comporten en parámetros esperados. La validación es asegurarse que cumplan los requerimientos de funcionamiento de una manera general.
2. No se puede asegurar que un código tenga las pruebas suficientes, siempre se tiene que tener en mente que un programa puede tener fallos a pesar de todos los intentos que pueden surgir en producción, lo mejor es analizar lo mejor posibles los usos de casos y generar pruebas unitarias, de rendimiento, etc. y tener lo mejor posible el diseño de este.
3. Son pruebas que aseguran que funcionan como se deben cada uno de los componentes del software.
4. Las pruebas unitarias prueban cada uno de los componentes de manera individual, las pruebas de integración siguen después de las pruebas unitarias y se aseguran que la interacción entre los componentes sea correcta.
5. QA es orientado al proceso, siendo una forma de prevención de fallos durante la construcción de un software, se diseñan y ejecutan métodos durante la creación del programa, QC es enfocado a corregir fallas ya cuando el programa está terminado, asegurando mejoras para el comportamiento final.
6. Las pruebas de rendimiento se efectúan para ver cómo se comportará el software cuando crezca y como se comportara en equipos con diferentes tipos de prestaciones.

Link al repositorio:

<https://github.com/rafaelhecr/innohub-test>