Descripción de Solución:

Se plantea una solución software aprovechando las capacidades de Spring-Boot para manejo de dependencias de especificación con H2 y Web.

La arquitectura MVC permite proponer un manejo de los componentes transaccionales solicitados.

@Controller

PersonController

@Service

PersonService : IPersonService

Servicios End Points REST:

* listRest
* addPersonRest
* processRest

Servicios Dao Interface

* list
* addPerson
* process

Person

JdbcTemplate

@Repository PersonDAO: IPerson

@RestController

PersonResource

jsp

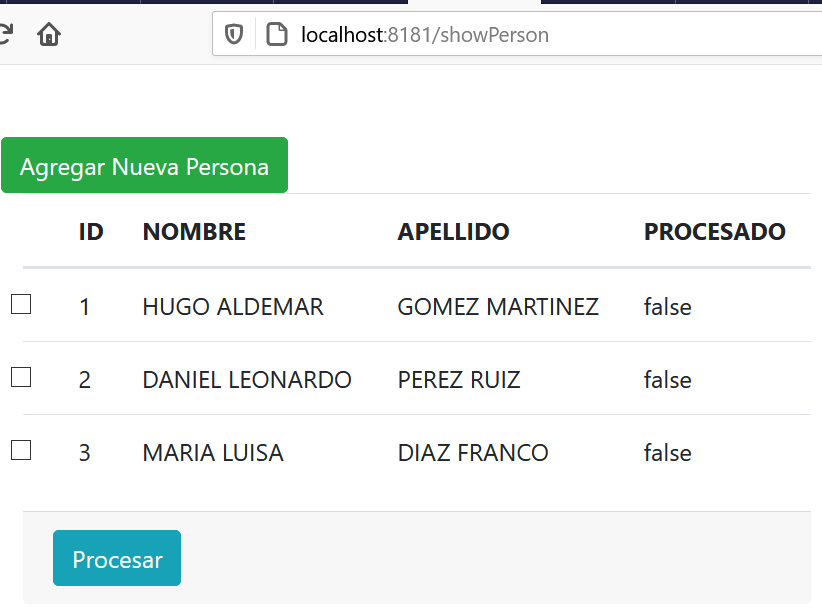
La propuesta arriba graficada, contempla la distribución de componentes propuestos para resolver el front-end (página que lista las personas y página que crea la persona) y el backend ( la lógica de negocio y los accesos a la interfaz de Datos que deben ser persistidos en BD H2).

Los componentes azules que harían parte del front-end básicamente muestran la implementación o consumo de los servicios Rest Expuestos por el @RestController del Backend. Este componente permitiría que los servicios fueran consumidos por cualquier app.

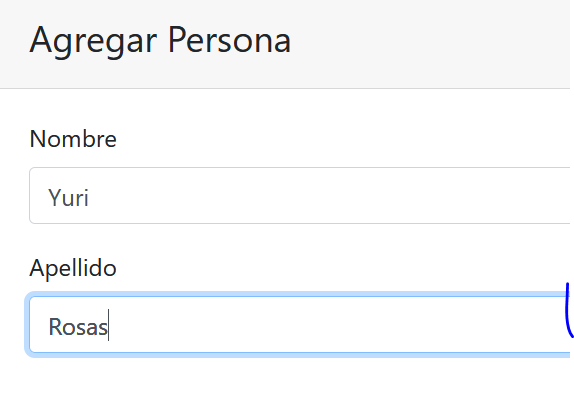
Los componentes verdes harían parte del back-end y ofuscan la lógica de negocio y tratamiento de datos.

Con el diseño propuesto, la idea es que se podrían crear dos (2) proyectos independientes con Azules y otro con verdes para ver el desacoplamiento dado que ambos compartirían la interfaz IPersonService con implementaciones particulares PersonServiceRest (Front) y PersonService (Back).

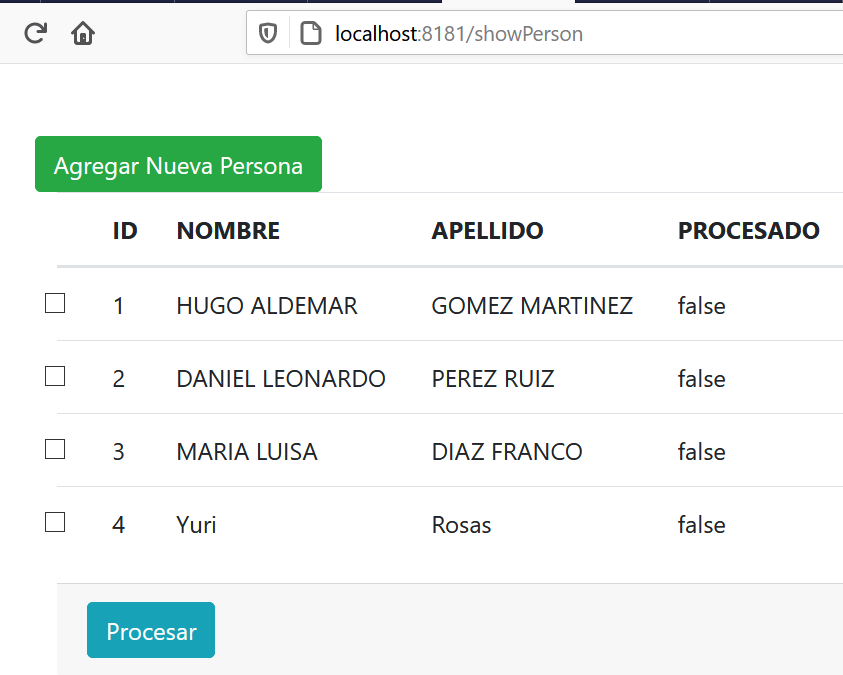
Funcional:



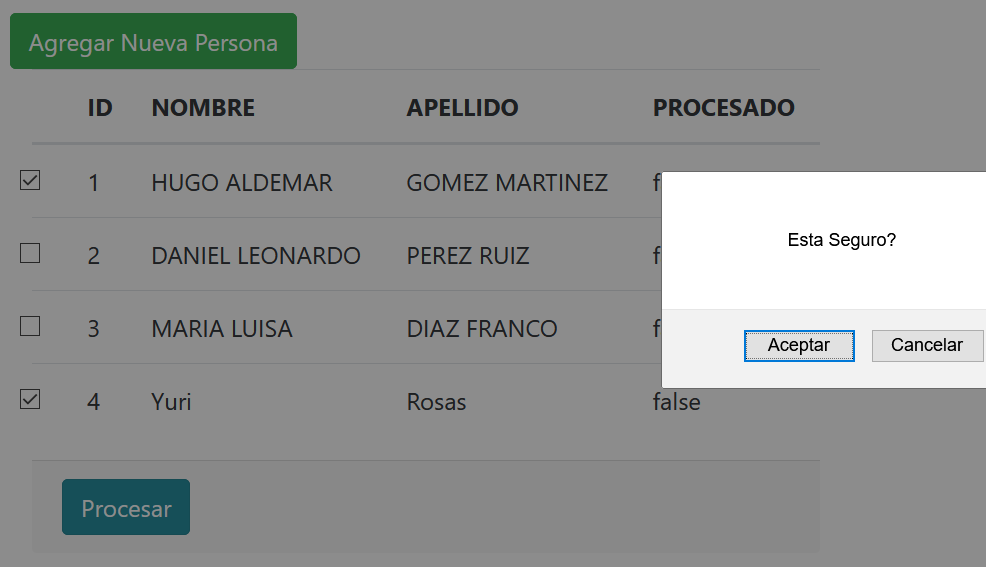
Click en Agregar Persona:



Click en Guardar:



Ahora, Procesamiento de Registros:



Resultado:

