

**[SIMULACRO 1] PRUEBA DE SALIDA FULL STACK JAVA**

Plan Formativo	Nivel de Dificultad:
Full Stack Java	Medio
<b>Nombre del proyecto:</b> Simulacro 1 para la prueba de salida del curso Full Stack Java	<b>Tema:</b> Generar una solución de negocio basada en una plataforma web construida bajo arquitectura JEE
<b>Objetivos del proyecto:</b>	<ul style="list-style-type: none"> <li>- Desarrollo de consultas a la base de datos</li> <li>- Crear algoritmos de acuerdo con requerimientos</li> <li>- Construir unidades de prueba</li> <li>- Crear una aplicación JEE</li> <li>- Implementar un servicio REST</li> </ul>

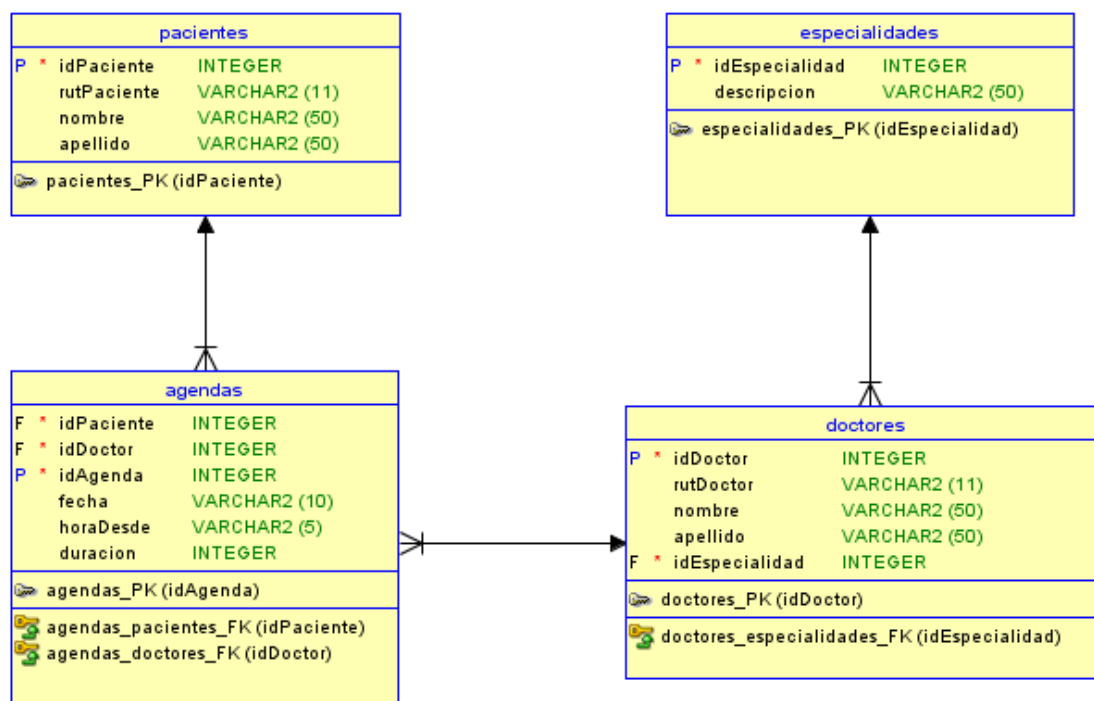
**Ejecución:** Individual

**Descripción del ejercicio**

**1.- Problema**

La Clínica “El Vacunazo” necesita llevar el control de las horas médicas que se solicitan a diario. Para ello, usted deberá desarrollar una solución que, basada en el modelo de datos descrito a continuación, satisfaga los requerimientos planteados más adelante.

El modelo de datos utilizado es el siguiente:



Las tablas anteriores y sus campos son los siguientes:

- **pacientes:** permite registrar a las personas que pueden agendar horas en la clínica. Considera un identificador de paciente, su RUT, su nombre y su apellido.
- **especialidades:** listado de especialidades médicas; se compone de un identificador y la descripción.
- **doctores:** en esta tabla se almacenan los doctores que prestan servicios a la clínica en diversas especialidades. Tiene un identificador, el RUT del doctor, el nombre y el apellido respectivos, y un identificador de la especialidad.
- **agendas:** cada registro representa una reserva de hora que hace un paciente asociándolo a un doctor específico. En ella se puede encontrar un identificador del registro, una referencia al identificador de doctor y del paciente respectivo, una fecha, una hora y una duración en minutos.

## 2.- Comandos iniciales de carga de datos

Para crear el modelo de datos en una base de datos Oracle 11g, ejecute el script que se adjunta al problema.

## 3.- Requerimientos a desarrollar

Desde la dirección de la clínica se solicita un sistema que pueda llevar un control de las horas agendadas; por lo tanto, le ha solicitado a usted que realice las siguientes tareas:

1. Realizar consultas a la base de datos.
2. Construir un algoritmo para el cálculo de tiempo agendado por día.
3. Construir una unidad de pruebas para verificar el algoritmo de cálculo del tiempo de horas agendadas.
4. Crear un reporte de horas agendadas.
5. Crear un servicio REST que despliegue la información del reporte de horas agendadas.

A continuación, se especifica con mayor detalle cada uno de los requerimientos:

### 3.1.- Realizar consultas a la base de datos

El directorio de la clínica le ha solicitado algunos reportes de información, razón por la cual el jefe del proyecto le ha encargado a usted que realice algunas consultas en la base de datos para la extracción de cierta información necesaria para el negocio, mientras que termina el desarrollo de la aplicación. Para esto, cree un package en su proyecto Java con nombre “consultas”, y agregue un archivo por cada una de ellas, identificando claramente de qué consulta se trata (ejm: Consulta-A.sql, Consulta-B.sql, ...etc).

- a) Listado de los doctores pertenecientes a la especialidad de Cardiología, indicando el apellido, nombre y RUT, y ordenados ascendentemente por el apellido y después por el nombre.
- b) Reporte que muestre la cantidad total de minutos agendados por cada día, desplegando la fecha de forma descendente, y asignando a la columna que representa la suma de minutos el nombre "TIEMPOTOTAL".
- c) Listado de los pacientes atendidos por el doctor Juan Manogrande, junto con el día y hora en que son atendidos. Debe estar ordenado ascendentemente por fecha y posteriormente por hora, desplegando del paciente su nombre y su apellido.
- d) Listado de todos los doctores registrados en sistema, indicando RUT del doctor, su nombre y su apellido, considerando además en otra columna la cantidad de citas agendadas. En caso de que el doctor no tenga horas agendadas, debe indicar un valor 0; además, la columna debe llevar por nombre "CANTIDADAGENDA". El resultado debe estar ordenado ascendentemente por el RUT del doctor.
- e) Listado de los pacientes (nombre, apellido y RUT) que se han atendido con doctores de la especialidad de Neurología. Este listado debe estar ordenado ascendentemente por el RUT del paciente.

### 3.2.- Construir un algoritmo para el cálculo de tiempo agendado por día

Un objetivo importante de este proyecto es solucionar el problema actual en el cual los administrativos de la clínica muchas veces no llevan el control correcto de los tiempos asociados a cada consulta. Como regla general, se sabe que la suma de los minutos de todas las consultas realizadas en un mismo día, independiente del doctor que la realice no debe superar los 300 minutos efectivos.

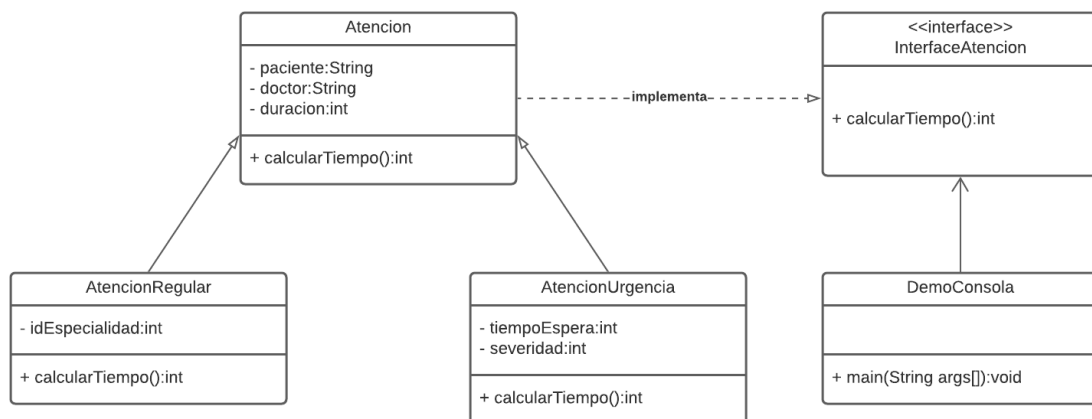
Además, existen dos tipos de prestaciones:

- **Regulares:** son las atenciones que se agendan y se realizan de manera normal. Por defecto una atención regular dura 15 minutos, excepto para las siguientes especialidades: en cardiología y neurología el tiempo de atención será de 25 minutos, mientras que para traumatología y gastroenterología la duración de la sesión será de 20 minutos.
- **De urgencia:** son atenciones que pueden requerirse en cualquier momento del día; adicional a los 15 minutos que se considera para cada atención, se debe sumar el tiempo de espera (se cuenta como parte de la atención), y un tiempo adicional dependiendo de la severidad: cuando la severidad es 1 (normal) se suman 5 minutos más, para severidad igual a 2 (grave) se suman 10 minutos, y para severidades de tipo 3 (muy grave) se suman 30 minutos.

Para dar solución a lo anterior, se decidió crear un algoritmo que realice la suma del tiempo asociado a todas las atenciones de un día, y que permita establecer si se cumple la regla del tiempo máximo de atención por día.

Esta es una idea que surgió recientemente pero que aún el jefe de proyectos no está del todo convencido de que dará el resultado esperado. Es por este motivo que el arquitecto ha decidido que lo mejor es construir una pieza de software extensible, con comportamiento polimórfico, para que a futuro se puedan incorporar nuevas lógicas de chequeo disminuyendo el impacto en el software.

Esta situación se puede representar a través del siguiente diagrama de clases:



Se le solicita que desarrolle un algoritmo de chequeo del tiempo de atención de pacientes con comportamiento polimórfico. Se pide entonces que implemente los siguientes elementos:

- Una interface de nombre `InterfaceAtencion`, en la que se defina un método de nombre `calcularTiempo()`.
- Una clase de nombre `Atencion`, que implemente la interface `InterfaceAtencion`.
- Las clases `AtencionRegular` y `AtencionUrgencia`, las cuales deben heredar desde la clase `Atencion`. En ellas se deben implementar los métodos `calcularTiempo()` respectivos, usando las reglas indicadas en los párrafos anteriores.
- El programa `DemoConsola`, el cual debe contener un método `main()` en el cual se cree manualmente (no es necesario solicitar por pantalla los datos) cinco atenciones de tipo regular y cinco atenciones de tipo Urgencia. Una vez creados estos objetos se debe obtener su duración invocando al método `calcularTiempo()` respectivo, y sumar dichas duraciones. Finalmente se debe indicar por medio de un mensaje por consola si esta suma de todas las atenciones superó o no el umbral diario establecido.

Considere el siguiente ejemplo de ejecución del programa a modo de referencia:

```
-----  
Demostración algoritmo cálculo tiempo  
-----  
  
Atención regular 1: Neurología, tiempo total 25 minutos  
Atención regular 2: Proctología, tiempo total 15 minutos  
Atención regular 3: Traumatología, tiempo total 20 minutos  
Atención regular 4: Gastroenterología, tiempo total 20 minutos  
Atención regular 5: Cardiología, tiempo total 25 minutos  
  
Atención de urgencia 1: Espera de 5 minutos, severidad tipo 3 (muy grave), tiempo total 50 minutos  
  
Atención de urgencia 1: Espera de 10 minutos, severidad tipo 1 (normal), tiempo total 30 minutos  
  
Atención de urgencia 1: Espera de 20 minutos, severidad tipo 2 (grave), tiempo total 45 minutos  
  
Tiempo total de atenciones en el día: 235 minutos  
  
Las atenciones del día NO superan el umbral establecido por la clínica
```

Genere dentro de su proyecto en eclipse un package con nombre representativo que tenga las clases mencionadas.

### 3.3.- Construir una unidad de pruebas

Construya una clase de pruebas en Java que permita verificar el correcto funcionamiento del algoritmo de cálculo de tiempo de atenciones, considerando al menos los siguientes tests:

- Tests que considere atenciones regulares, con distinto tipo de especialidad (2 casos).
- Tests que considere atenciones de urgencia, con distintos tiempos de espera y severidad (2 casos).

### 3.4.- Crear un reporte de horas agendadas

Para mejorar la gestión de las agendas de consultas médicas, se requiere crear una página web dinámica que despliegue el reporte de horas agendadas, tal como se detalla en la siguiente imagen mock-up.

Mozilla

← → ↻

## Listado de Reservas

Especialidad:

Paciente	Doctor (a)	Especialidad	Fecha	Hora
Arthur Sanders	Emerson Lucas	Cardiología	28/08/20	18:15
Ochovia Beloseno	Jamal Hinton	Psiquiatría	28/08/20	11:30
Ochovia Beloseno	Josefa Amasabubis	Gastroenterología	24/08/20	17:30
Quinn Shelton	Alika Nixon	Traumatología	24/08/20	19:45
Quinn Shelton	Chanda Sanford	Psiquiatría	24/08/20	18:00
Rodolfo Nalgonius	Juan Manogrande	Proctología	24/08/20	12:00
Rodolfo Nalgonius	Tamekah Bond	Ginecología	25/08/20	09:00
Valerie Panzoni	Peter Hale	Neurología	31/08/20	10:45
Valerie Panzoni	Rebeca Güatuza	Proctología	24/08/20	14:15

Se pide:

- Desplegar el listado de especialidades en el elemento de tipo select, ordenado alfabéticamente, con valores que provengan de la base de datos.
- Al abrir la página por primera vez, se debe desplegar el listado de horas agendadas (de acuerdo con el detalle especificado en la imagen mock-up) sin ningún tipo de filtro, y ordenando ascendentemente por nombre del paciente y nombre del doctor.
- Al presionar el botón Buscar, se deben desplegar solo aquellos registros correspondientes a la especialidad seleccionada. Si no se seleccionó ninguna especialidad, se deben desplegar todos los registros. En ambos casos se debe respetar el orden de los campos.

Para realizar el requerimiento, el arquitecto le señala lo siguiente:

- Utilizar jsp y taglibs jstl para el despliegue de la vista (u otra tecnología de vista)
- Utilizar bootstrap para los elementos
- Que los elementos se ajusten a distintos tamaños de pantalla

### 3.5.- Crear un servicio REST que despliegue la información del reporte de horas agendadas

Disponibilice un servicio REST que permita obtener la misma información del Reporte de Horas Agendadas. Recuerde que el servicio podría recibir como parámetro el identificador de

especialidad (en caso de que desee desplegar todos los registros, el identificador debe ser igual a cero).

#### Requerimientos

- El sistema debe ser construido utilizando el framework Spring MVC, conectándose a una base de datos Oracle 11g express.
- En las tablas no existen campos autoincrementales.
- La revisión del problema se realizará en base al modelo antes planteado; no se permite agregar, modificar o quitar campos del modelo.

#### Contribuciones

#### Requerimientos de los participantes

<b>Conocimientos previos</b> <ul style="list-style-type: none"><li>• HTML</li><li>• CSS</li><li>• Javascript</li><li>• Responsividad</li><li>• Java Enterprise Edition</li><li>• Spring Framework</li><li>• Oracle 11g express edition</li><li>• Servicios Rest</li></ul>	<b>Actitudes para el trabajo</b> <ul style="list-style-type: none"><li>• Proactividad</li><li>• Aplicar casos anteriores en contextos similares</li><li>• Uso del tiempo</li><li>• Efectividad en la solución planteada</li></ul>	<b>Valores</b>  Tiempo de resolución.  Iniciativa
<b>Objetivo General de Aprendizaje</b>	Desarrollar una plataforma que permita mostrar los registros provenientes de una búsqueda, junto con un servicio REST de obtención de datos.	
<b>Objetivos particulares</b>	<ul style="list-style-type: none"><li>- Creación de sitios web responsivos</li><li>- Conectar un sitio a una base de datos</li><li>- Desarrollar una solución en base a un framework</li></ul>	
<b>Duración del proyecto</b>	Cuatro horas	
<b>Tips o listado de Preguntas Guía</b>		

Productos a obtener durante la realización del proyecto
<ul style="list-style-type: none"><li>- Un sitio web compuesto por dos o más vistas</li><li>- Debe obtener datos en formato JSON desde un servicio Rest, y cargarlos en una base de datos</li></ul>
Especificaciones de desempeño
Deberá realizar la actividad según requerimientos técnicos en un plazo máximo de 4 horas
Cronograma de actividades
Sugerencias bibliográficas para la investigación