

## PROYECTO: PUCP Private Cloud Orchestrator

La Pontificia Universidad Católica del Perú cuenta con una serie de servidores DELL Power Edge R720, los cuales planea usar para implementar una nube privada, de manera que pueda ser utilizada para el despliegue de topologías de máquinas virtuales (Virtual Machines) conectadas dinámicamente en capa 2. Dada la cantidad de máquinas virtuales a provisionar, y el dinamismo, se requiere un **sistema automatizado** que ayude al administrador de la infraestructura con el despliegue de los recursos.

Se le pide que implemente la primera versión de este sistema, la cual no tendrá un entorno gráfico. Su equipo de trabajo debe diseñar una solución que facilite el aprovisionamiento dinámico y minimice la posibilidad de error humano. Los requerimientos de esta solución son:

- **R1 [14 pts]:** Controlar la creación y el borrado de slices de máquinas virtuales en una infraestructura de nube genérica, ya sea privada (Linux cluster, OpenStack, etc.) o pública (AWS, GCP, etc.).
- **R1B [15 pts]:** Proveer una interfaz de usuario que permita a un usuario gestionar el ciclo de vida de sus slices (creación, edición, borrado, etc.).
- **R1C [21 pts]:** Gestionar el despliegue/aprovisionamiento de slices sobre la Infraestructura de nube deseada. Responsable porque los diferentes elementos del slice sean creados en el orden adecuado y en el lugar deseado y que la creación del slice sea completa (no se quede a medias).
- **R2 [24 pts]:** Soportar la creación/borrado de slices sobre un cluster de servidores Linux.
- **R3 [21 pts]:** Soportar la creación/borrado de slices sobre un sistema OpenStack.
- **R4 [28 pts]:** Ejecutar el "VM Placement" (mapeo de VM a servidor físico) tomando en cuenta los recursos disponibles y permitiendo elegir la "zona de disponibilidad" a usar.
- **R5 [21 pts]:** Proveer soporte de Networking capas 2 y 3, permitiendo definir reglas de ciberseguridad

Su grupo diseñará e implementará una solución que cubra tantos requerimientos como les sea posible. En función a la calidad de su solución, recibirá un puntaje total (nota) que será repartido entre sus miembros de la manera que los miembros del grupo decidan.

### Criterios de evaluación

Descripción	Criterios
<b>R1</b> Controlar la creación y el borrado de slices de VMs en una infraestructura de nube privada. <b>(14 pts)</b>	<ul style="list-style-type: none"> <li>• Diseño independiente de la capa de infraestructura (cluster de servidores Linux, OpenStack, AWS, etc.).</li> <li>• Debe tener un módulo para cada uno de los requerimientos subsiguientes:               <ul style="list-style-type: none"> <li>- Debe poder almacenar la información de forma persistente.</li> <li>- Debe permitir la comunicación en red entre módulos.</li> <li>- Debe soportar la creación concurrente (multiusuario) de slices, es decir, manejo de colas.</li> <li>- Debe soportar la creación de logs con la información necesaria para hacer troubleshooting de cada módulo.</li> </ul> </li> </ul>
<b>R1B</b> Interfaz de usuario que gestiona el ciclo de vida de slices <b>(15 pts)</b>	<ul style="list-style-type: none"> <li>• Permitir que múltiples usuarios usen la infraestructura de forma concurrente.</li> <li>• Permitir seleccionar slices con topologías predefinidas: lineal, malla, árbol, anillo, y bus.</li> <li>• Permitir <u>editar slices existentes</u> agregando nodos y enlaces.</li> <li>• Permitir especificar la configuración (capacidad) de cada VM a crear.</li> <li>• Listar los slices existentes.</li> <li>• Listar el consumo de recursos del sistema.</li> <li>• Permitir borrar slices previamente creados.</li> <li>• Debe permitir importar imágenes de VMs.</li> <li>• Proveer tokens/credenciales para que el dueño del slice pueda acceder a la consola virtual de cada VM.</li> </ul>
<b>R1C</b> Gestor de despliegue de slices ("Slice Manager") <b>(21 pts)</b>	<ul style="list-style-type: none"> <li>• Expone los APIs del sistema. Puede ser un sólo módulo o varios.</li> <li>• La comunicación del resto de módulos debe pasar por él.</li> <li>• Debe validar que el usuario esté autenticado y tenga autorización de realizar la acción solicitada.</li> <li>• Debe validar que el body de las solicitudes tengan la sintaxis correcta.</li> <li>• Modela los slices y sus elementos (uso de OOP).</li> <li>• Crea los elementos en el orden correcto y lidia con errores y con conflictos por el uso de recursos.</li> </ul>
<b>R2</b> Soporte de cluster de servidores <b>(24 pts)</b>	<ul style="list-style-type: none"> <li>• Orquesta VM sobre servidores corriendo alguna versión de linux.</li> <li>• Ahorro de almacenamiento: debe borrar las imágenes de las VMs que ya no se usen (en los servidores).</li> <li>• Proveer acceso a la consola virtual de cada VM.</li> </ul>
<b>R3</b> Soporte de OpenStack <b>(21 pts)</b>	<ul style="list-style-type: none"> <li>• Usar el OpenStack API.</li> <li>• Implementar al menos Keystone, Horizon, Nova, Neutron (ML2 plugin), y Glance.</li> <li>• Debe permitir ver cada slice creada, por separado, en Horizon.</li> </ul>

<b>R4</b> VM Placement <b>(28 pts)</b>	<ul style="list-style-type: none"><li>• Asignar las máquinas virtuales a los servidores de manera que:<ul style="list-style-type: none"><li>- El consumo de recursos no exceda un nivel máximo de sobre-aprovisionamiento (parámetro de configuración).</li><li>- La carga está balanceada entre los servidores.</li></ul></li><li>• Listar los recursos (servidores) disponibles.</li><li>• Permitir al usuario definir zonas de disponibilidad.</li><li>• Permitir al usuario elegir en qué zona de disponibilidad se desplegará un slice.</li></ul>
<b>R5</b> Networking y Seguridad <b>(21 pts)</b>	<ul style="list-style-type: none"><li>• Proveer salida a Internet y reglas de seguridad a las VMs que lo requieran.</li><li>• Debe permitir acceso desde Internet a las máquinas virtuales que lo requieran (equivalente a tener una IP pública).</li><li>• Usar redes provider (VLANs) en lugar de self-service (tuneles).</li><li>• Limitar los paquetes de broadcast a aquellos puertos/servidores que tienen VMs que pertenezcan a la subred respectiva (ya sea usando Netcong, Yang, OpenFlow, etc.).</li></ul>

## Plan de acción

Sem	Fecha	Actividades
0-1	11 Mar / 24 Mar	<ul style="list-style-type: none"> <li>- Repaso Linux, Bash, Python</li> <li>- Revisión del diplomado en Cloud Computing &amp; otra bibliografía.</li> </ul>
2	25 Mar / 31 Mar	<ul style="list-style-type: none"> <li>- Repaso de Linux Networking y Linux Process Management</li> <li>- <b>Lab 1: Bash &amp; Python scripting + Linux Process Management</b></li> <li>- Primera discusión del proyecto y sus alcances.</li> </ul>
3	01 Abr / 07 Abr	<ul style="list-style-type: none"> <li>- Revisión del diplomado en Cloud Computing &amp; otra bibliografía</li> <li>- Revisión de tesis de Ruben Cordova y Antonio Merino</li> <li>- Lecturas adicionales provistas por "coach" y los miembros.</li> <li>- Redacción de requerimientos del proyecto</li> </ul>
4	08 Abr / 14 Abr	<ul style="list-style-type: none"> <li>- Repaso de Tecnologías de Virtualización</li> <li>- <b>Lab 2: Aprovisionamiento de un slice de VMs en un solo host.</b></li> <li>- Presentar, como parte del reporte final del lab 2: <ul style="list-style-type: none"> <li>1. concepto de operación y/o casos de uso</li> <li>2. descripción de entidades y roles en su solución</li> <li>3. bosquejo de arquitectura (módulos e interacciones entre ellos).</li> </ul> </li> </ul>
5	15 Abr / 21 Abr	<ul style="list-style-type: none"> <li>- Repaso de Tecnologías de Virtualización (cont.)</li> <li>- Desarrollo del borrador de arquitectura de solución: módulos y sus funciones.</li> </ul>
6	22 Abr / 28 Abr	<ul style="list-style-type: none"> <li>- Repaso de Tecnologías de Virtualización (cont.)</li> <li>- <b>Lab 3: Avances de la arquitectura y user interface</b></li> <li>- El documento y PPT con desarrollo de arquitectura debe subirse a PAIDEIA 1 día antes del Lab 3.</li> </ul>
7	29 Abr / 05 May	<ul style="list-style-type: none"> <li>- Repaso de Tecnologías de Virtualización (cont.)</li> <li>- Desarrollo del High level software design del User Interface (UI) de la aplicación y el esquema de la base de datos.</li> </ul>
8	06 May / 12 May	<ul style="list-style-type: none"> <li>- Repaso del problema de "VM Placement"</li> <li>- <b>Lab 4: Aprovisionamiento de un slice de VMs en un cluster</b></li> <li>- El documento con HLD UI + BD + Scheduler debe subirse a paideia 1 día antes del Lab 4 (parte de nota del lab)</li> </ul>
9	13 May / 19 May	<ul style="list-style-type: none"> <li>- <b>1<sup>RA</sup> PRESENTACIÓN PROYECTO: LUNES 13 MAY. 2024, 3:00 PM</b></li> <li>- PRESENTACIÓN DE ARQUITECTURA Y DISEÑO DE LA INTERFAZ DE USUARIO (UI). La UI debe interactuar con <u>algunos</u> APIs (hardcoded) y la arquitectura debe incluir el diagrama de clases, BD y la lógica del scheduler. Debe indicar los requerimientos que su arquitectura busca satisfacer (y que por lo tanto orientan su diseño).</li> </ul>
10	20 May / 26 May	<ul style="list-style-type: none"> <li>- Repaso del problema de "VM Placement" (cont.)</li> <li>- Desarrollo del High level software design de los módulos restantes</li> </ul>

11	27 May / 02 Jun	<ul style="list-style-type: none"> <li>- Repaso de OpenStack</li> <li>- Implementación del módulo que orquesta VMs en Linux Cluster.</li> <li>- <b>Lab 5: Análisis y despliegue de OpenStack</b></li> </ul>
12	03 Jun / 09 Jun	<ul style="list-style-type: none"> <li>- Repaso de OpenStack (cont.)</li> <li>- Revisión de tesis de Gabriel Gamero.</li> <li>- Definición con coaches del plan de pruebas.</li> <li>- Unit testing: la aplicación imprime el output de <u>todos</u> los comandos que enviaría al resto de módulos pendientes de implementación</li> </ul>
13	10 Jun / 16 Jun	<ul style="list-style-type: none"> <li>- Repaso de Contenedores</li> <li>- <b>Lab 6:</b> <ul style="list-style-type: none"> <li>o Informe Previo: Aprovisionamiento de VMs usando: (i) la CLI de OpenStack , y (ii) la API de OpenStack.</li> <li>o Experiencia de Lab: presentación de la implementación, el plan de pruebas y los resultados de las pruebas realizadas hasta el momento.</li> </ul> </li> <li>- 1 día antes del Lab 6 deben subir el plan de pruebas.</li> </ul>
14	17 Jun / 23 Jun	<ul style="list-style-type: none"> <li>- Repaso de Contenedores (cont.)</li> <li>- Implementación del módulo que orquesta VMs en Linux Cluster (cont.)</li> </ul>
15	24 Jun / 30 Jun	<ul style="list-style-type: none"> <li>- El mercado de Cloud Computing</li> <li>- Implementación del módulo que controla el switch de datos (OFS)</li> <li>- <b>Lab 7: Implementación de un módulo del proyecto del curso, incluye UNIT Testing y reporte de pruebas.</b></li> </ul>
16	01 Jul / 07 Jul	<ul style="list-style-type: none"> <li>- <b>PRESENTACIÓN FINAL LUNES 1 Julio 2024, 3:00 PM</b></li> </ul>

## Entregables

<b>Primera Entrega</b>  <b>POR DEFINIR</b>	1. Arquitectura de la solución (documento PDF, requerimiento R1) incluyendo APIs y/o IPC (arquitectura del resto de requerimientos). Debe indicar cómo la arquitectura permite satisfacer sus requerimientos.
	2. High level software design (documento PDF) del UI y manager del orquestador (requerimientos R1B y R1C)
	3. Demo de interfaz de usuario (UI), mostrando el output intermedio que se mandaría al driver que controle la infraestructura subyacente (OpenStack, AWS, Cluster de servidores, etc.).
	4. Código de la interfaz de usuario (UI)
	5. Copia de presentación (powerpoint, PDF, etc.)
<b>Entrega Final</b> <b>POR DEFINIR</b>	6. High level software design (documento PDF) de todos los módulos
	7. Código de la versión final de su prototipo.
	8. Demostración de funciones
	9. Reporte de Pruebas
	10. Copia de presentación (powerpoint, PDF, etc.)

## Calificación del curso

La nota final del curso se calculará con la siguiente fórmula:

$$N_f = 0.4P_b + 0.3Ex_1 + 0.3Ex_2$$

Donde

Pb : Promedio de los 7 laboratorios.

Ex1 : Nota asignada a la primera entrega del proyecto (09-Oct.).

Ex2 : Nota asignada a la segunda (y final) entrega del proyecto (27-Nov.).

Además, tenga en cuenta que el laboratorio 5 consistirá en la implementación de uno de los módulos de su proyecto, así que el peso del proyecto en la nota final es aún mayor.

**Nota:** la implementación tiene el doble del peso que la arquitectura.

A continuación se muestra una tabla con la distribución de puntajes

	Requerimientos							
	R1	R1B	R1C	R2	R3	R4	R5	
Arquitectura	14	5	7	8	7	10	7	58
Implementación	0	10	14	16	14	18	14	86
	14	15	21	24	21	28	21	<b>144</b>

**Nota de EX1:** corresponde a los requerimientos R1, R1B y R1C (arquitectura + implementación), más la componente de arquitectura de los módulos R2, R3, y R5 (excepto "R4: VM placement"). Total 72 pts.

**Nota de EX2:** corresponde al componente de implementación de los requerimientos R2-R5 más el componente de diseño/arquitectura del requerimiento R4. Total 72 pts.

En ambas presentaciones, basta obtener 60 de los 72 puntos para ser asignado el puntaje máximo.