

Tarea 2

Bases de Datos

*“Cryptomonedas: P-Coin **II**”*

Diego Altamirano diego.altamiranot@sansano.usm.cl	Javier Mendoza javier.mendoza@sansano.usm.cl
Felipe Quintanilla felipe.quintanilla@sansano.usm.cl	Kevin Reyes kevin.reyesv@sansano.usm.cl
Rodrigo Rojas rodrigo.rojasmu@sansano.usm.cl	Cecilia Reyes reyes@inf.utfsm.cl

15 de Mayo, 2021

1. Introducción

Tras la excelente entrega en las solicitudes de su tarea anterior, los administradores de la universidad han decidido continuar el desarrollo del proyecto, comenzando una nueva etapa de producción: Un sitio web.

El Departamento de Informática les pide a los alumnos de Bases de Datos que apliquen sus conocimientos en consultas SQL y generen una plataforma que sea capaz de soportar las distintas operaciones que requiere este sistema. Para ello, se solicitará que trabajen con tecnologías de desarrollo web, conexiones a bases de datos, y logren coordinar su trabajo en equipo para generar un diseño consistente y funcional, que consiga mantener el gran prestigio de nuestro departamento.



2. Instrucciones de Entrega

Para esta entrega, su objetivo será diseñar y generar un sitio web que sirva como plataforma base para trabajar con las criptomonedas entregadas en la primera iteración de esta tarea. La **plantilla** adjunta a esta tarea viene con una versión simple de las vistas que se les solicitarán, las cuales carecen de implementación lógica. Se les recomienda enormemente **analizar todos los archivos contenidos en la plantilla**, pues muchos de ellos presentan comentarios indicando en qué partes hace falta implementar la lógica solicitada.

2.1. Modelo de Datos

Antes del desarrollo en sí, es necesario atender el siguiente requisito:

- Desde el mismo modelo de la entrega anterior, deben adaptar sus atributos o tablas para que distingan **privilegios de administrador**, es decir, que de alguna forma su base de datos pueda diferenciar entre un usuario que es **Admin** y uno que no lo es.
- Por lo tanto, deberán analizar el modelo e implementar los cambios que sean pertinentes para satisfacer el requisito.
- Existen múltiples formas válidas de implementar el requisito, por lo que deberán **justificar su elección, señalando ventajas y desventajas**.

2.2. Requisitos del Sistema

2.2.1. Sesiones

Una gran cualidad de la plataforma es que debe sostener el uso de **sesiones**, es decir, debe permitir que un Usuario ingrese sus datos, mantenga una sesión y pueda salir cuando así lo desee. Específicamente se necesita que:

- Un usuario pueda Iniciar Sesión, ingresando sus datos (correo y contraseña) en una interfaz común entregada por la plataforma.
- Cuando un Usuario ingrese sesión, el sitio debe determinar autónomamente si el Usuario es Admin o **no**.
- Dentro de su sesión, el Usuario, independiente de su nivel de acceso (Admin o no), debe tener una pestaña o interfaz que le permita ver su perfil (información de su cuenta).
- Además, debe ser capaz de cerrar sesión desde cualquier vista de la plataforma.

2.2.2. Interfaces

Como núcleo de funcionamiento, el sistema debe soportar interfaces y/o formularios que cumplan:

- Una **Página Principal**, comúnmente conocida como *index* o *frontpage*, la cual sirva como una bienvenida o "Primera impresión" de la plataforma, donde un Usuario que no la conoce pueda entender en qué consiste, sus desarrolladores, qué servicios entrega, etc.
 - Si no se especifica una URL particular de la plataforma (sólo la IP o dominio), esta debe ser la vista a la que llegue el Usuario.

- Si se intenta entrar a una página de forma indebida (sin tener una sesión válida o teniendo los privilegios incorrectos), se debe redirigir al usuario a la página principal.
- Una interfaz de **Sign In/Sign Up** (Registro e inicio de sesión); donde un Usuario podrá iniciar sesión, entregando su correo y contraseña, o un visitante podrá registrarse en la página web, ingresando todos los datos necesarios para ello.
 - Se debe poder acceder a estas vistas sólo si no se ha iniciado sesión. En caso contrario, debe redirigirse a la página principal.
- Un **CRUD** para los Usuarios del sistema, donde se puedan crear nuevos usuarios, ver todos los Usuarios registrados en este, modificar sus atributos o borrarlos.
 - Esta vista sólo puede ser accedida por un **Administrador**.
- Una interfaz de **Perfil**, la cual muestra los Datos personales al Usuario.
 - Cualquier Usuario puede acceder a esta vista, siempre y cuando ya este registrado.
- Una interfaz de **Billetera**, donde se muestren las Monedas que tiene el Usuario y su valor actual en el mercado.
 - Esta interfaz debe indicar el **valor actual** en USD (Dólares Estadounidenses). Considere que todas las referencias a valores se harán sobre esta moneda.
 - La interfaz debe incluir una **conversión total**, donde se indique al Usuario cuantos USD podría obtener si convirtiera todas sus monedas actuales a Dólares (USD).
 - La interfaz no debe mostrar monedas que el Usuario no tiene, es decir, no debe mostrar una moneda con 0 pertenencia de esta.
 - Esta vista estará disponible para los Usuarios comunes (no administradores).

3. Condiciones de Entrega

3.1. Informe

Dentro del mundo del desarrollo, nada está 100 % estandarizado, y las ideas e implementaciones varían de desarrollador en desarrollador. Es por esto que, independiente de cómo o qué implementen en su entrega, les solicitamos que adjunten a esta un **Informe**, el cual detallará todas sus consideraciones, suposiciones, detalles de uso y notas respecto a la plataforma. Como mínimo se solicita que el informe:

- Debe incluir los **datos personales de cada alumno** de su grupo, junto con su número.
- *(Opcional) Si lo desean, pueden adjudicarse un “Nombre de Grupo” y utilizarlo como marca o empresa desarrolladora de la página.*
- Debe incluir **todos los supuestos y consideraciones** que uds. tomaron en cuenta al desarrollar la entrega. Además, si se agregó algún archivo que no venía en la plantilla, debe indicarse qué se agregó, dónde y con qué propósito (Por ejemplo: *Se agregó foto.png a la carpeta img para mostrarse en la página principal*).
- Debe señalar **cómo se manejó el Modelo** para permitir sesiones de Administrador y Usuario, mostrando al menos una ventaja y desventaja de la forma en que lo plantearon.
- Debe indicar, si es que los hay, qué archivos de la plantilla fueron modificados. Esto **no considera** archivos tanto `.html` y `.php` con comentarios indicando que hagan cambios.
 - Expliciten también si las modificaciones fueron de funcionalidad o simplemente estéticas (ejemplo: uso de Bootstrap para hacerlas más estéticas).
- En una sección final, solicitamos que informen acerca de sus **dificultades** con esta implementación; en concreto, que puedan reportar cuáles tareas fueron las más difíciles (o largas) en su desarrollo. Puede variar desde instalación de tecnologías, ajuste de modelo, proceso creativo, desarrollo, etc.
- En esta misma última sección, solicitamos que reporten un estimado de cuánto **tiempo** les tomó el desarrollo de la tarea completa (valores por persona), buscando incluir todo momento que gastaron en ella.
 - *Estimadores Recomendados: Análisis del Enunciado, Modificaciones y ajustes al Modelo, Diseño de plataforma, Desarrollo de Plataforma, Pruebas Finales.*

El informe debe venir en formato **PDF**, e incluir las secciones detalladas anteriormente.

3.2. Implementación

Respecto a la implementación de los requisitos, se solicita:

- Utilicen la **Plantilla** entregada como base de su tarea. Esta ya tiene todos los archivos que necesitarán para desarrollar las funcionalidades de forma ordenada y eficaz.
- El código debe estar **ordenado**, es decir, indentado, estructurado y lo más comentado posible.

- **Importante:** Se debe hacer un buen uso de SQL, sin buscar que la lógica del sitio ejecute operaciones costosas que debieran ser resueltas por las consultas a la base de datos. Por ejemplo, se considerará un **mal uso** el que, en vez de utilizar un JOIN, itere sobre una consulta con un *for* para hacer una consulta sobre cada elemento de la primera (estilo pythoniano).
 - Nota: Esto no implica que no deban utilizar ciclos (como *for*) en su código, nos referimos particularmente a reemplazar las consultas por procesamiento de la plataforma.
- Se debe mantener la estructura de los archivos lo más ordenado posible. En caso de agregar archivos nuevos a la plantilla, deben seguir la misma línea que los ya existentes.
- *Bonus 5pts: Utilizar Bootstrap u otra librería de desarrollo para que las vistas sean estéticas y consistentes.*

4. Condiciones Generales

- La fecha de entrega es el **Domingo 6 de Junio a las 23:59PM**.
 - Se aplicará un descuento lineal para tareas atrasadas hasta las **11:59 AM** del día siguiente.
- Las tecnologías a utilizar para el desarrollo de la plataforma son:
 - **Apache 2.4:** Montar plataforma con sus vistas (*Front-End*)
 - **PHP 7.3+:** Para manejar la lógica del negocio, es decir, el código detrás de la página
 - **PostgreSQL 12:** Para el manejo de la base de datos asociada al sistema (*Back-End*)
- Su entrega debe incluir el **informe** (nombre de archivo **InformeGrupoX**) y la **carpeta** de su implementación (nombre de carpeta **src**), además de un **dump** de su base de datos modificada (nombre de dump **DumpGrupoX**). Todo esto en un comprimible de nombre **Tarea2-GrupoX.tar.gz**, donde X es el número de su grupo.
 - *PeaZip* es una buena herramienta para generar comprimibles distintos a *.rar*
 - Recordar que en el gitbook del curso hay una sección sobre cómo hacer dumps de la base de datos en Pgadmin4
- **Bonus 5pts.:** Para el desarrollo de su entrega, esta permitido el uso de **Git** para el control de versiones y trabajo colaborativo. De hacer esto, en su comprimible reemplacen la **carpeta** de implementación por un archivo *repoGrupoX.txt* con el **link a su repositorio**.
 - El repositorio no puede ser privado, o como mínimo, debe ser accesible por nosotros Ayudantes.
 - Esta permitido que el README de su repositorio sea el **Informe** solicitado para esta entrega.
 - Los puntos extra se otorgarán sólo si se evidencia un buen uso del Git, lo que abarca commits de distintos integrantes del grupo a través del tiempo, notas de implementación, y similares.
- La entrega de los archivos es por el medio Aula, sólo uno de los integrantes del grupo debe realizar la entrega.

5. Recomendaciones

- Aún el gran número de tecnologías en el actual oficio de desarrollo de software, es común que la instalación de tecnologías sea un paso difícil de realizar. Sin embargo, les recomendamos que, ante cualquier fallo o impedimento, **busquen en internet si alguien ha tenido el mismo problema**. Muchas veces horas de peleas se ahorran con un milagroso post en [Stackoverflow](#) del 2014.
- Esta tarea requiere en parte, el uso de creatividad y diseño por parte de ustedes. Sin embargo, esta es una excelente oportunidad para crear su propia página web y decorarla como a uds grupo les parezca mejor, implicando paletas de colores, organización de interfaces y/o contenido visual para el Usuario, temáticas, etc.
- A pesar de que no es necesario, Bootstrap es una excelente herramienta para quien quiere añadir estética a su página. Es sencillo de instalar y mucho más fácil de usar que otras tecnologías (y les servirá mucho para ramos posteriores).
- Parte del trabajo es programar en PHP, lo que implica nuevos entornos y comandos a utilizar. Ante cualquier duda o incógnita que tengan sobre el lenguaje, les recomendamos con mucha fuerza revisar W3Schools. Es la biblia del informático (e incluye un apartado de Bootstrap!).
- Finalmente, consejo general de desarrollador: Si quieres saber cómo hacer X con Y tecnología, googlea “How to do X with Y” (o en español si les complica el Inglés) y revisa los resultados.