

Fin primer Sprint

Grupo 1

touchFood

Proyecto de Software

Índice

Introducción/desarrollo.....	1
Aguirre Iván.....	2
Armanno Matías.....	6
Cabado Leonel.....	10
García Leonardo.....	12
Giménez Martín.....	14
Leguizamón Gonzalo.....	16
Navarro Gabriel.....	19



Introducción

En el siguiente informe se especifican los detalles que ocurrieron a lo largo del primer sprint por de los parte los integrantes del Grupo 1. Estos son: Aguirre Iván, Armanno Matías, Cabado Leonel, García Leonardo, Giménez Martín, Leguizamón Gonzalo y Navarro Gabriel.

Desarrollo

El fin de este informe es especificar las tareas que realizó cada miembro del grupo. Para esto cada uno realizó una parte del mismo detallando el rol que ejecutó en este primer sprint, los que problemas tuvo, como los resolvió y que enseñanza le dejó.



Aguirre Ivan Gabriel

ROL: Front-End

Como desarrollador front-end de TouchFood, mis primeros momentos los pase ideando y pensando como quería diseñar las pantallas de la app.

Para tener una buena referencia, en conjunto con mi compañero Leo García (Cani), me compartió MaterializeCss que es una librería de CSS, (<http://materializecss.com/>).

Allí me puse a investigar más para ver de qué se trataba, como trabajaba y funcionaba.

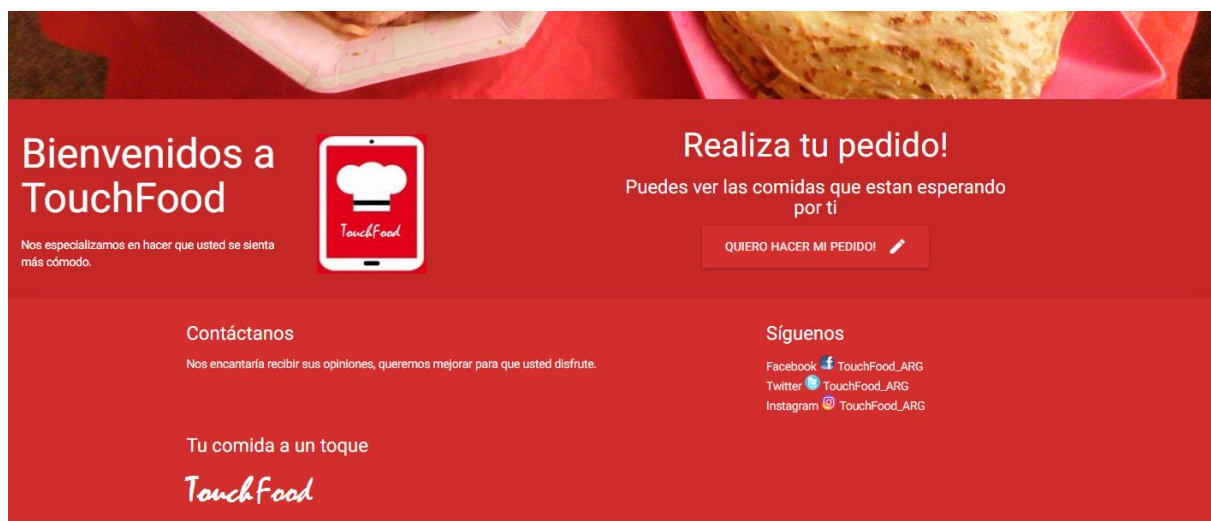
Finalmente fue el complemento que utilizamos para el front de la aplicación, a veces ayudado por Bootstrap (<http://getbootstrap.com/>) pero muy pocas veces, fue utilizado para algunos botones, no mucho más.

Al empezar con el diseño, tuve problemas primarios, los cuales fueron:

- ¿De qué color va a ser?
- ¿Qué tipo de letras va a tener?
- ¿Cuál iba a ser nuestro logo?

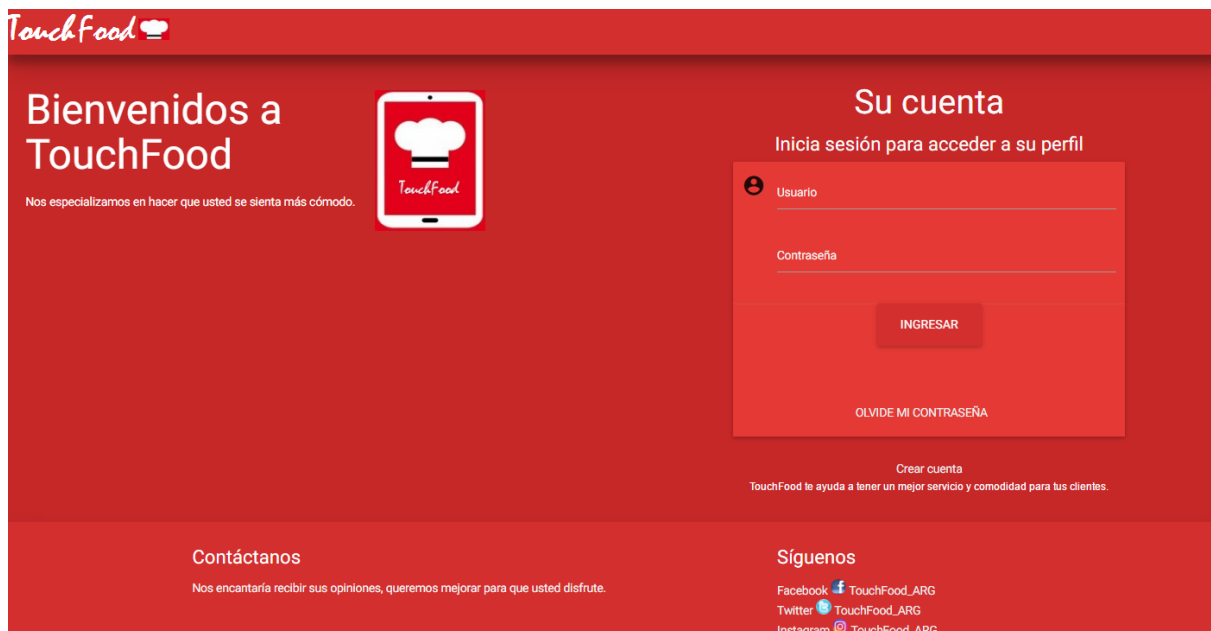
Luego de resolver estos problemas, investigando y preguntando a mi grupo, pudimos llegar a un acuerdo.

Así fue como quedo nuestra página de Bienvenida (de parte cliente):





En la pantalla de “Bienvenida_Administrador” que sería la principal, pero para el administrador, quedo así:



En esta experiencia no hubo muchos problemas, ya que se tomo el template de la pantalla “Bienvenida” (cliente).

Uno de los problemas que surgió fue alinear la parte del login al lado de “Bienvenidos a TouchFood” pero investigando un poco, viendo unos tutoriales en internet (videos de YouTube), y leyendo el material que proporciona Materializecss pude solucionarlo rápidamente.

Otro problema que ocurrió fue el diseño del menú, como íbamos a hacerlo.

Uno de los problemas que tuve en este diseño, fue que al hacer un foreach y traer los datos de las categorías de la base de datos, estos datos estaban escritos todos en minúscula.

Mi compañero de front, Leo, me paso un código (ucwords) el cual podía hacer que las primeras letras estén en mayúscula, y así fue.

```
</div>
<li class="collection-header red darken-2"></li>
@foreach($categorias_comidas as $categoria_comidas)
<li class="collection-item red darken-1"><div> {{ ucwords($categoria_comidas->categoria) }}<a href="comidas/{{ $categoria_comidas->categoria }}" class="g
@endforeach
>
```

Otro problema que se presento fue que, en los carteles de cada categoría, debía ir el nombre justamente de la categoría, y al insertarle una foto, este nombre no se podía ver correctamente.

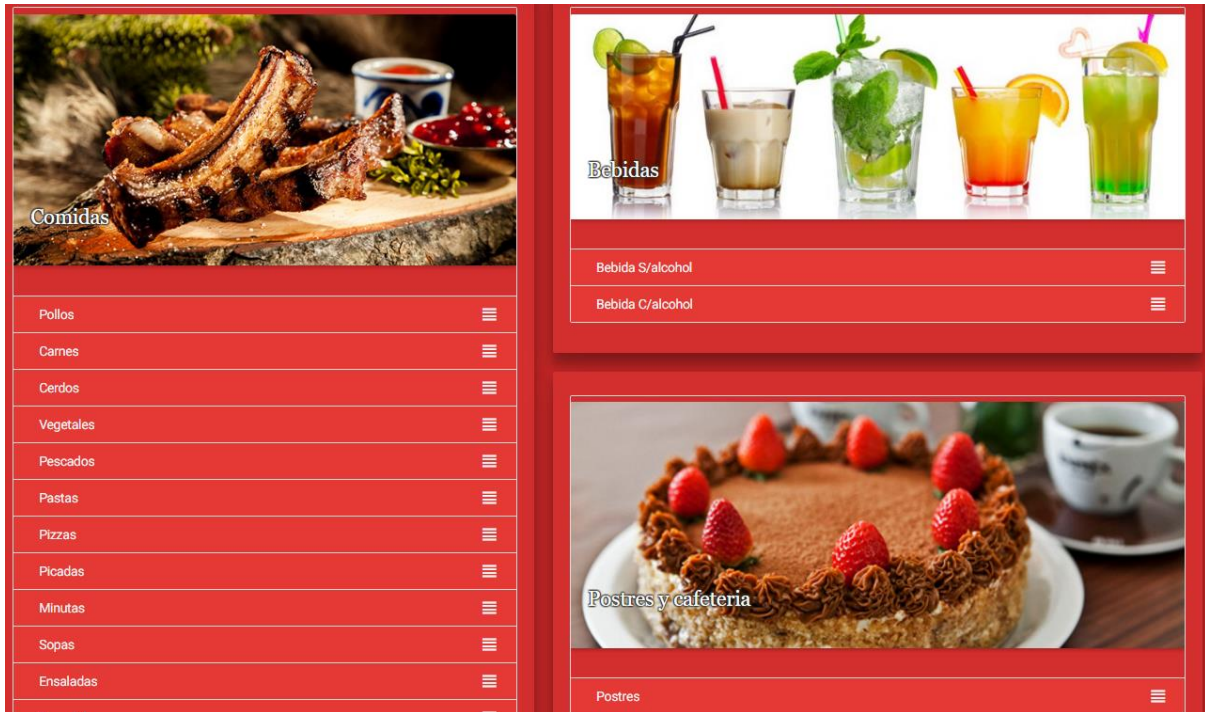
Este problema fue resuelto gracias a una función de CSS (stroke).

```
.stroke {  
  text-align: center;  
  color: white;  
  font-family: Georgia, serif;  
  font-size: 40px;  
  letter-spacing: 10;  
  text-shadow: -1px -1px 1px #333, 1px -1px 1px #333, -1px 1px 1px #333, 1px 1px 1px #333;  
}
```

Y así quedó gráficamente.



Finalmente se hizo de esta forma.



Conclusión:

Es la primera vez que trabajo en este rol de front-end, pero debo decir que me sentí muy cómodo hasta ahora, y me gustó mucho lo que hice, pase mucho tiempo viendo como me gustaría que quedaría, preguntando a mis compañeros que les parecía y manifestando mis opiniones, para que quede siempre un poco mejor que antes.

Creo que un buen diseño de front-end es muy importante para la aplicación, por eso tanto énfasis, es lo que más llama la atención al público.



Armando Matías Nahuel
Rol: Analista Funcional

En un principio del proyecto mi idea era formar parte en el rol de back-end, pero al ser un grupo tan numeroso, casi todos apuntábamos a lo mismo y nadie pretendía estar en rol de analista. Es por eso que decidí tomarlo (con poco entusiasmo) y comencé a investigar las labores que estos hacían y las zonas que debían ocupar para aportar al grupo.

Como solamente tuve experiencia desarrollando y no en este sector, simplemente contaba con los conocimientos teóricos y casos de trabajos pedagógico que no tenían un fin productivo y ni siquiera entorno real. Pero me basé en ellos para analizar, junto con el equipo, los requerimientos funcionales que la aplicación demandaba y las zonas que debíamos focalizarnos para comenzar a desarrollar.

Las tareas que dejan un rastro a la vista son las de desarrollo de documentación. En este caso comencé con la tarea de realizar el SRS (Especificación de Requerimiento de Software) y para ello decidí utilizar como base el formato que ofrece el estándar IEEE 830 y ver ejemplos de documentos reales para casos de software existentes como los siguientes:

<https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>

http://es.slideshare.net/Juan_Tapias/formato-ieee830srs-lleño

<https://es.scribd.com/document/325070555/ejemplo-de-especificacion-de-requerimientos-para-sesion-9-pdf>

<http://www.juntadeandalucia.es/servicios/madeja/contenido/libro-pautas/182>

Gracias a esto el SRS ya se encuentra casi finalizado con las especificaciones de los requerimientos funcionales que elegimos para este primer sprint. Solamente queda ultimar algunos detalles de cara a la presentación y aceptación por parte del grupo.

Hablando con un conocido ajeno al proyecto que estudia una carrera enfocada al análisis de mercado y temas relacionados, me comentó casos de aplicaciones similares (no idénticas, luego lo detallo) que no llegaron a nuestro país. Es por eso me interioricé a recopilar información e investigué de casos en el mercado. No encontré aplicaciones que sean exactamente iguales pero estas fueron las que más se relacionaban:



<http://mobileworldcapital.com/es/285/>

<http://www.emprendedores.es/ideas-de-negocio/el-camarero>

<http://www.libremercado.com/2013-11-24/nace-una-aplicacion-para-pedir-sin-esperar-al-camarero-1276504929/>

<http://www.opentable.com/start/home>

Con toda esta información me lancé a desarrollar un informe sobre objetivos de negocio, estudio de mercado, técnicos, económicos y financieros para consultar y en un futuro sirva como referencia queremos darle al proyecto una vuelta de tuerca y dejar de mirarlo como una tarea universitaria y enfocarnos a su explosión en el propio mercado. En estos momentos el informe se encuentra en la fase inicial de su desarrollo.

A continuación, como ejemplo nuestro capturas de determinados fragmentos del SRS y algunos requerimientos funcionales que se manejaron para este Sprint

TouchFood				
SRS - Especificación de requerimiento funcionales				
Código	Nombre	Fecha	Grado de necesidad	
RF-2	Categorías	25-ago-16	Esencial	
Descripción	Creación de las categorías de base de datos			
Entrada	Fuente	Salida	Destino	Restricciones
Consultas	Comidas y usuarios	Categorías de las comidas y usuarios	Base de Datos	Cada tipo de comida debe tener su categoría
Proceso	Se desarrollará las categorías de las comidas en las bases de datos con sus respectivos relaciones, las mismas serán Pollo, Carnes, Cerdo, Vegetales, Pescado, Pastas, Pizzas, Picadas, Minutas, Sopas, Ensaladas y Especiales. Además de los usuarios del sistema			
Efecto colateral	Permite traer por medio de controladores las comidas con las que va a contar el sistema y el tipo de usuarios que habrá			



TouchFood				
SRS - Especificación de requerimiento funcionales				
Código	Nombre	Fecha	Grado de necesidad	
RF-6	Interfaz de bienvenida	08-sep-16	Esencial	
Descripción	Creación de interfaz de bienvenida de la aplicación			
Entrada	Fuente	Salida	Destino	Restricciones
Get de la pagina	Llamada a controlador	Interfaz de bienvenida	Interacción con usuario	Debe poseer las características previamente establecidas conforme a la forma que debe poseer
Proceso	Se mostrará una página de bienvenida que propondrá un entorno que cumpla con las pautas previamente establecidas, las líneas de contacto por medio de redes sociales y el login para acceder al sistema			
Efecto colateral	Entorno y comunicación ágil y simple con el usuario			

Introducción

1 Introducción

El presente documento contiene las Especificaciones de Requisitos de Software para la aplicación de reserva de mesas, gestión de pedidos de platos y pagos automáticos para un restaurante. La estructura del mismo está basada en las directrices referidas en el estándar IEEE 830-1998 Recommended Practice for Software Requirements Specifications.

1.1 Propósito

El objetivo de esta especificación es definir de manera clara y precisa las funcionalidades y restricciones que tendrá el sistema que se desea construir, y va dirigida al equipo de desarrollo de software y a las personas que harán uso del sistema terminado. Este documento será un medio de comunicación entre cada uno de los roles implicados en el desarrollo de software y por lo mismo está sujeto a revisiones, tanto de los desarrolladores como de los usuarios, hasta obtener su aprobación. En cuanto esto ocurra el documento funcionara como base para informar las características del proyecto.

1.2 Ambito del Sistema

El sistema se conocerá con el nombre de "touchFood", y se encargará de ofrecer servicios de autogestión a clientes de restaurantes, lo cual beneficiará tanto a la organización del restaurante como a la gestión de ventas, reduciendo costos y tiempos de espera.

1.3 Personal Involucrado

Nombre	Aguirre Iván
--------	--------------



Conclusión

Mi mayor dificultad era relacionar las herramientas que me había brindado la facultad a lo largo de la carrera (como por ejemplo con materias como Ingeniería de Software I y Metodologías de Programación II) con un caso de trabajo real y lo resolví viendo ejemplos y recopilando información que me sirvan como guía de comienzo.

Como antes mencioné, no estaba muy convencido del rol que me había tocado. Pero me di cuenta que como tuve que investigar todas las herramientas de desarrollo para tener conocimientos del proyecto, estoy atento a todas las funcionalidades que se van sumando e integrando, formo parte al igual de todos a las charlas donde intentamos resolver problemas que se nos presentan, no era real que siendo Analista Funcional quedaría excluido de la parte de desarrollo.

Debido a esto, conocí un sector que quizás yo menospreciaba, pero en verdad me es de agrado realizar ya que tengo que ejecutar tareas de investigación, análisis, propuesta de ideas que aporten y conocimiento de casos para mejorar mi labor y contribuir de forma práctica al grupo. Esto me lleva a la idea de que quizás, en un futuro, pueda desarrollarme profesionalmente en esta labor.



Cabado Franco Leonel

Rol: Back-end

En los primeros pasos en el framework, intente utilizar vagrant y homestead para llevar acabo el proyecto laravel. Vagrant es una herramienta que crea un entorno de desarrollo basado en máquinas virtuales, através de una interfaz crea servidores perfectamente configurados e independientes del sistema operativo del desarrollador, lo que significa que estos no deben realizar ninguna configuración para levantar el entorno de trabajo. En cuanto a homestead, es una imagen de sistema operativo que se crea con vagrant que viene ya configurado con un paquete de software tales como Ubuntu, PHP, HHVM, Nginx, MySQL, PostgreSQL, NodeJS, Redis y algunas librerías como Memcached, LaravelEnvoy, Beanstalkd.

Luego por ciertas complicaciones, decidimos con el equipo trabajar con xampp. Mi rol específico fue tareas de backend, realice el modelo ORM (mapeo de objeto-relacional) llamado Eloquent incluido en laravel, codifique los modelos de las clases correspondientes a cada tabla de la base de datos en cuestión especificando además las relaciones que existían entre estas tablas.

Además realice la configuración de la conexión de la base de datos con los eloquent, aporte para la implementación del controlador encargado de consultar valores a la base de datos para luego enviarlos a la vistas.

Por último estuve a cargo del realizar el login lo que se me complico bastante ya que la documentación de laravel no explicitaba muchos aportes con respecto a esto, sumándole que no contaba con mucha experiencia con el framework, ya que nosotros necesitabas logearnos con usuario y no con e-mail. Investigando encontré una manera muy acotada:

```
public function store(LoginRequest $request)
{
    $oferentes = oferente::all();

    foreach ($oferentes as $ofer) {
        if ($ofer->contraseña==$request['contraseña'] and $ofer->razon_social==$request['razon_social']) {
            return Redirect::to("test");
        }
    }
    return Redirect::to("menu/categorias");
}
```

En este script me traigo todos los registros de la tabla oferente de la base de datos y luego los recorro con un foreach en busca de los campos Contraseña y razon



social, a su vez utilizo una vista de formulario la cual me recopila en la variable &request lo ingresado por el usuario y luego con un if comparo si coinciden los datos obtenidos de la tabla con los del request, de coincidir o no utilizo Redirect para dirigirme a una vista en particular. Esta opcion fue acotada ya que en ella no se tiene en cuenta ni las protecciones de las rutas ni la generacion de sesiones y por lo tanto el proyecto quedaba muy vulnerable.

Luego encontre otra forma de hacer según la documentacion de laravel, y fue con la utilizacion de metodo attempt, este metodo recibe un array con los campos a autenticar y devuelve true o false dependiendo dicha autenticacion.

```
    return $this->redirect($request->route('login'));
}

/**
 * Handle the incoming request.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(LoginRequest $request)
{
    if(Auth::attempt(['razon_social'=>$request['usuario'], 'password' =>$request['password']])){
        return Redirect::to("menu");
    }
    //dd($request['usuario']);
    //dd('razon_social');
    return Redirect::to("ver");
}
```

En este caso tuve el problema de que el método attempt siempre devolvía false por más de que los campos coincidían, analizando un poco el framework encontramos que como en laravel se encuentran métodos hardcoded el método attempt solo respondía correctamente si se utilizaba todo el bloque de autenticación que proponía laravel.

Conclusión:

Si bien al principio costo un poco entender el funcionamiento de algunos aspectos del framework creo que en poco tiempo aprendi cosas bastante interesantes del mismo. Igualmente todavía me quedan muchas cosas por que practicar e investigar por ejemplo la parte del logeo para autenticarse con usuario, la cual quedo para el próximo SPRINT.



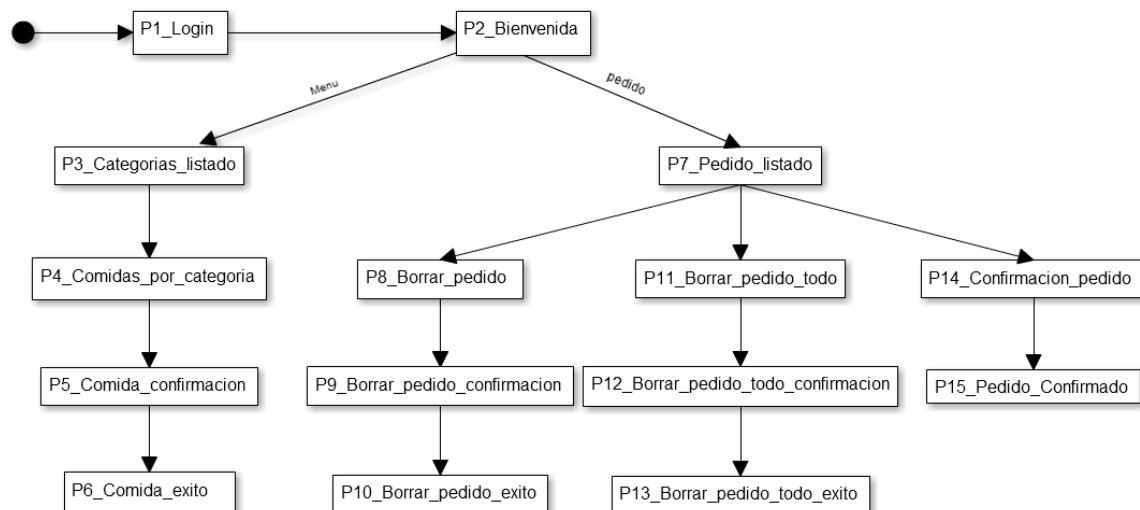
Garcia Leonardo

Rol: Front-end (SM)

Al inicio del proyecto, se buscó una distribución de roles equitativa según nuestras áreas de conocimientos. Al principio me había inclinado por el desarrollo front end, pero luego por falta de un Scrum Master en el equipo de desarrollo decidí tomar dicho rol.

Comencé investigando acerca de los distintos frameworks y el modelo MVC y proponiendo el aprovechamiento de un framework que esté vigente en el mercado decidimos optar por Laravel. Desde este punto me puse a investigar sobre esta herramienta, como implementa el modelo-vista-controlador, el sistema de migraciones a la base de datos, el sistema de plantillas Blade, entre otros tópicos, para luego ir guiando a mis compañeros, ayudarlos cuando les surja alguna duda y realizar una distribución de tareas.

De manera simultánea, comenzamos con el encargado de base de datos a empezar a pensar el sistema pensando en un paradigma arquitectónico para atacar los problemas por separados y concluir en que dicho sistema debe tener dos posiciones o partes, la del usuario que va a comer al restaurant y la del personal de que lo atenderá. Se decidió atacar por la primera parte y pensar la base de datos con las relaciones propias de los restaurantes (menús, comidas, categorías, pedidos, etc.). Se pensó el siguiente flujo de negocio, sin tener en cuenta la parte de pagos aun:



Luego se comenzó a plantear un diseño general de la página. Lo cual recurrí a una librería de CSS llamada Materialize, que me pareció que sería la indicada

Sprint 1



debido a que posee una curva de aprendizaje muy baja que se adecua a los tiempos que tenemos, la cual le puse a disposición a mi compañero Iván Aguirre para que comience a desarrollar.

Para tener un mejor orden de trabajo cargue las tareas al Pivotal Tracker, designadas para cada miembro del equipo.

También contribuí con el back-end resolviendo la pantalla de categorías, la cual aprovechando las posibilidades de ruteo de Laravel, tomamos la categoría que clickeamos y la coloca en la URL, la cual la recibe un controlador que espera la URL/categoría_elegida y según eso realiza la consulta a la base de datos. Luego de esto, decidí realizar un paso más y hacer un listado de estas comidas con paginación, la cual será cambiada por un scroll infinito, utilizando la librería jqueryinfiniteScroll, que está en etapa de prueba debido a la poca cantidad de comidas en la base de datos.

Por último luego de un exhaustivo trabajo para realizar un login personalizado, el cual no tuvo éxito, siguiendo la documentación de Laravel, implemente un sistema de autenticación con la tupla mail-contraseña, la cual no es adecuada a nuestro sistema porque la autenticación es de mesas, pero provisoriamente esta es la solución que encontramos.

Conclusión

Como desenlace de este primer Sprint me llevo una buena experiencia debido a que lo más difícil, que es la adaptación al framework, se logró exitosamente por parte de todo el equipo. Lo cual nos resulta beneficioso y nos incrementa la propiedad colectiva del código. El equipo de desarrollo ha cumplido completamente las expectativas que esperaba, se puede observar siempre la incorporación de nuevas ideas al proyecto por parte de todos y eso demuestra una participación y compromiso que se verá reflejado en el producto final. Si bien no pudimos realizar todas las pantallas que habíamos propuestos, considero que para el próximo sprint esto mejorará y se verán reflejados aúnmas resultados productivos.



Giménez Martín

Rol: Back-end

En el inicio del proyecto fui el encargado de buscar información sobre un Gestor de tareas y el repositorio para alojar el proyecto. Me decidí por crear/configurar el proyecto en Pivotal Tracker, proponiendo esta App(la había utilizado en Metodologías de Programación II) porque me parece una herramienta muy completa, además como está orientada a la gestión de proyectos ágiles, por ejemplo para proyectos con Scrum nos era más que indicada ya que esta era la metodología que elegimos.

De esta herramienta podemos destacar algunas ventajas tales como:

- El proyecto si está en Public es gratis y no tiene ningún tipo de limitación.
- Está en la nube. Con lo que no tenemos que instalar nada en nuestros ordenadores, todo lo haremos a través del navegador.
- Se centra en algo muy interesante que es la gestión de la pila de producto, y métricas como la velocidad.

También cree el repositorio en GitHub. Me incliné por GitHub por sobre Bitbucket ya que esta es una plataforma especialmente diseñada para el trabajo colaborativo. Además ofrece varias pequeñas herramientas en línea muy útiles para el trabajo en equipo. Entre ellas, podemos destacar:

- Un sistema de seguimiento de problemas, que al estilo del clásico sistema de tickets en las empresas, permite a los miembros de tu equipo (o a cualquier usuario de GitHub si tu repositorio es público) abrir un ticket detallando un problema que tenga con tu software o una sugerencia que desee hacer al mismo.
- Una herramienta de revisión de código, donde se pueden añadir anotaciones en cualquier punto de un fichero (ej: “Esto es mejor que lo extraigamos a un nuevo fichero”), y debatir sobre determinados cambios realizados en un archivo específico.
- Un visor de ramas donde se pueden comparar los progresos realizados en las distintas ramas de nuestro repositorio.

Vale aclarar que busque que el repositorio y el gestor de tareas se puedan integrar.

Sprint 1



Del repositorio tuve que volver a crear otro porque el primero no funciono (desconozco lo que sucedió).

Además traté de ir indicando como utilizar estas 2 herramientas a aquellos compañeros que no las habían utilizado.

Luego tuve diversos inconvenientes en la configuración del proyecto en Laravel, incluso llegué a cambiar de distribución (de Xubuntu a Linux Mint) creyendo que en las diversas soluciones pude haber tocado algo de más, ya sea en la instalación de varias versiones de PHP o MySQL, etc.

Esto hizo que me fuera retrasando con respecto a los demás compañeros del equipo. Igualmente en clases preguntaba e iba viendo los avances que se iban logrando ya sea en la parte de Backend o front. También propuse ideas para la realización del Logo.

Investigué sobre los distintos procesos de Autenticación en Laravel, viendo la documentación de Laravel, leyendo tutoriales y realizando diferentes pruebas de los mismos.

Conclusión

Las diversas complicaciones que tuve en la instalación/configuración del proyecto en Laravel me hicieron perder tiempo y retrasarme en el aprendizaje del framework y todavía no estoy a la par de mis compañeros. Igualmente estoy logrando mucha experiencia en un grupo de desarrollo de muchas integrantes, aprendiendo, además, sobre nuevas herramientas. En fin es una experiencia muy grata hasta el momento desde la idea de la Aplicación, la cual que me parece bastante interesante, hasta la retroalimentación positiva que se da en el grupo.

Leguizamon Gonzalo

Rol: Base de Datos

Como dba de touchfood, los primeros momentos los pase pensando como sería el modelo de negocio y como funcionaria la aplicación para adaptarse a dicho modelo, es decir los conocimientos que tienen que ver con el dominio de los restaurantes.

Para lograr cierto conocimiento de dicho dominio, y poder diseñar una base de datos que se acopló a ello, visite varias páginas web sobre restaurantes, para asesorarme del tema de los menús (ya sea categoría y comidas como también presentación grafica). Entre ellas, Casa manolo



<http://www.restaurantecasamanolo.com/carta-y-menus-restaurant-pamplona>

Luego de diseñada y creado el script de la base de datos, tuve varias dudas en como concebir a los pedidos, si hacerlos con una sola factura en donde irían todos los platos agregados(realidad), o si crear una factura por plato, bebida o postre pedido, luego de charlarlo con el docente se decidió llevar a cabo la primera opción, para la cual implemente una vista en la base que traiga todos los pedidos (comidas, bebidas y postres) de las diferentes tablas y los asocie directamente a una sola factura.

Sprint 1



Imagen de la vista:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	num_factura	int(11)			No	0	
2	mesa	int(11)			No	Ninguna	
3	estado_factura	varchar(25)			No	Ninguna	
4	plato	int(11)			No	Ninguna	
5	hora_plato	time			No	Ninguna	
6	poste	int(11)			No	Ninguna	
7	hora_postre	time			No	Ninguna	
8	bebida	int(11)			No	Ninguna	
9	hora_bebida	time			No	Ninguna	

Otro problema el cual nos trajo varias dudas fue el login, ya que laravel hardcodea sus clases para hacerlo con una tabla predefinida User, por lo cual, cuando intentábamos autenticar con nuestra tabla que contenía los usuarios(oferente), el login no funcionaba.

Gracias a la ayuda del profesor y a horas leyendo documentación del framework, pudimos hacer funcionarlarlo con dicha tabla Users la cual trae para loguear por defecto Email y password.

Finalmente luego de hacer los scripts, el tester Navarro Gabriel detecto que tenia un bug en los campos de tiempo de cocción, y tiempo de preparación de las comidas y postres, estos son de tipo(Date), el cual posee el formato hh:mm:ss y yo había introducido los valores en la sección de los segundos, en vez de introducirlos en la seccion de los minutos. Por lo tanto tuve que arreglar dichos campos.

Sprint 1



Asimismo también participe configurando el archivo .env para la conexión de la base al programa y las consultas por categorías de comidas en la aplicación a través de los modelos.

Conclusión

Si bien, decidí hacer la parte de bases de datos por qué es lo que más me gusta y donde mayor conocimiento tengo aprendí como utilizar los modelos mapeadores de objetos, y diferentes cosas como a hashear contraseñas, utilizar vistas, etc.

En fin creo que el diseño de base le aporta mucho a la aplicación y determina su funcionamiento dentro de las restricciones de dominio previamente evaluadas, y es una experiencia grata y satisfactoria.



Navarro Gabriel

Rol: Tester

Comprobación en Base de Datos.

Una de las primeras labores realizadas por parte del grupo fue la creación de la Base de Datos, por ello, también fue una de las primeras cosas para probar. Para realizar dicha actividad se analizó la base de datos realizando diversas consultas para chequear datos: consultas a tablas para ver la información y por otro lado, modificaciones e inserciones. Las pruebas resultaron satisfactorias, salvo por el pequeño error en el campo que tenían en común las tablas de “comidas”, donde la columna de “tiempo de preparación” estaba mal expresada: el tiempo se estaba brindando solamente en segundos, cuando debía ser en minutos. Esto fue rápidamente corregido por el DBA.

Luego de estas pruebas y consultas contra la base de datos, se procedió a testear, de manera más visual, que las diversas URL a las que se podía acceder desde el proyecto trajeran correctamente la información hacia el browser: por ejemplo, si uno accedía a la URL **`http://localhost/Proyecto_Touch/public/menu/comidas/pollos`**

Debía ver los datos traídos desde la BD respecto a la tabla “pollos”, con sus columnas de ‘descripción’ y ‘precio’, por ejemplo. Con ello se probó visualmente que esto se realizaba correctamente, y se procedió a insertar algunos nuevos elementos para corroborar que posteriormente se iban a poder añadir datos.

Pruebas Unitarias (PHPUnit).

Para las siguientes pruebas se procedió a realizar pruebas unitarias, de la mano de PHPUnit. Dicho framework ya viene instalado por default con Laravel.

Las pruebas consistieron en corroborar que las URL configuradas por el archivo “routes.php”, llamaran correctamente a sus correspondientes controladores, y estos procedieran a llamar correctamente los datos y la vista. Por ejemplo, al acceder a la URL **`http://localhost/Proyecto_Touch/public/menu/categorias`** se debe mostrar cierta información que puede ser chequeada visualmente, para ello se creó el siguiente test, el cuál fue creado por “Consola”, mediante el comando `php artisan make:test MenuTest`. En esta clase Test, se ubican los diversos TestCase para chequear los diferentes controladores, URLs y vistas que debe mostrar nuestra aplicación. El test de categorías es el siguiente:



```
TestTest.php x Find Results x .env x TestController.php x
1 <?php
2
3 use Illuminate\Foundation\Testing\WithoutMiddleware;
4 use Illuminate\Foundation\Testing\DatabaseMigrations;
5 use Illuminate\Foundation\Testing\DatabaseTransactions;
6
7 class MenuTest extends TestCase
8 {
9     /**
10      * Test para probar el contenido de los menús al ser traídos de la BD.
11      *
12      * @return void
13      */
14
15     public function testMenuCategorias()
16     {
17         $this->visit('/menu/categorias')
18             ->see('Comidas');
19     }
20 }
```

Se comprueba que al acceder a la URL /menú/categorías, se deba mostrar el mensaje de “Comidas” en alguna parte de la página. Este mecanismo se utilizó para corroborar las demás pantallas, chequeando que se traiga correctamente alguna información característica de esa página, sabiendo así que la consulta y la pantalla muestran información correcta.

A continuación, se muestran más de los TestCase hechos:

```
public function testMenuPollos()
{
    $this->visit('/menu/comidas/pollos')
        ->see('pollo con fritas');
}

public function testMenuCarnes()
{
    $this->visit('/menu/comidas/carnes')
        ->see('asado de tira');
}

public function testMenuCerdos()
{
    $this->visit('/menu/comidas/cerdos')
        ->see('lechón asado para 2');
}
```




En los TestCases se sigue la misma lógica, entrar a la URL y corroborar que se muestra información correcta:

```
public function testMenuPicadas()
{
    $this->visit('/menu/comidas/picadas')
        ->see('tablita caliente');
}

public function testMenuMinutas()
{
    $this->visit('/menu/comidas/minutas')
        ->see('hamburguesa');
}

public function testMenuSopas()
{
    $this->visit('/menu/comidas/sopas')
        ->see('sopa clasica');
}
```

Resultado de ejecución de algunos test:

```
C:\Windows\system32\cmd.exe

C:\xampp\htdocs\Proyecto_Touch - copia>phpunit
PHPUnit 4.8.27 by Sebastian Bergmann and contributors.

.....
Time: 3.15 seconds, Memory: 12.25MB
OK (16 tests, 32 assertions)
C:\xampp\htdocs\Proyecto_Touch - copia>
```



Problemas.

El principal problema encontrado fue la inexperiencia y escaso conocimiento respecto al framework de Test de PHP. El poco conocimiento era respecto a las pruebas unitarias en general. Este inconveniente fue solucionado a medias, ya que aún falta ahondar más en el tema, intentando crear más pruebas unitarias a medida que surjan nuevas funcionalidades, más código en sí. De todas formas, con la ayuda de Internet se fue obteniendo la ayuda necesaria hasta el momento.

Por otro lado, otro inconveniente, pero de índole más personal, fue el hecho de haber comenzado a trabajar full time en la zona de Puerto Madero / Retiro. Esto afectó fuertemente mi participación en el proyecto, debido a que no podía brindarle toda la atención y energía posible. Sin embargo, poco a poco me voy acomodando y espero poder estar de mejor manera para el siguiente sprint.

Conclusión

Para concluir, se puede decir que testear los proyectos es fundamental para el buen funcionamiento de lo que va a realizarse. Si bien no se pudo llegar a realizar test de muchas cosas, por falta de conocimiento y tiempo más que nada, se llegó a realizar pruebas interesantes que complementaron de mejor manera al proyecto y ayudaron al mismo. Quizá el testing no es el área donde más me gustaría desarrollarme profesionalmente, tener conocimientos sobre ello ayudará a que sea mejor profesional.